



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Projekto „Arduino Nano GUI integracija su Pimatic“

Dokumentacija

Studijų modulis: T120B139 Aplinkos kompiuterizacija ir protingos sistemos

Projekto autorius:
Kristupas Cilcius, IFB-2

Prie projekto idėjos vystymo prisidėjo:
prof. Egidijus Kazanavičius

Kaunas, 2024

TURINYS

1. Projekto aprašymas.....	3
1.1. Pagrindinė Projekto idėja	3
1.2. Projekto funkcionalumas.....	3
1.3. Naudoti įrankiai Projekto kūrimui.....	3
1.4. Kūrimo procesas.....	4
1.5. Projekto schema	4
1.6. Iššūkiai ir idėjos ateičiai	4
2. Kaip naudotis projektu?.....	4
3. GitHub	6
Išvados.....	6

1. PROJEKTO APRAŠYMAS

1.1. PAGRINDINĖ PROJEKTO IDĖJA

Projektas kuriamas būsimiems „Aplinkos kompiuterizacija ir protingos sistemos“ studentam, kurie galės laisva valia naudotis šiuo projektu ir/ar keisti programos Processing4.3 kodą pagal savo poreikius. Šis projektas buvo sugalvotas tam, kad studentams būtų lengviau įsivaizduoti, kaip veikia Pimatic aplinkoje siunčiamos homeduino protokolo komandos į Vhduino.c kodą, ir tai jiems padėtų lengviau atlikti laboratorinius darbus bei suprasti, kaip vyksta aplinkos kompiuterizacijos ir protingų sistemų integracija su mikrovaldikliais (šiuo atveju Arduino Nano).

Kadangi Vhduino.c kodas skaito komandas, ateinančias homeduino protokolu ir tuo pačiu simuliuoja tam tikrus jutiklių paramtrus, kaip temperatūra ir drėgmė (šia simuliuojama DHT sensoriaus parametrai) ir juos atitinkamai siunčia atgal į Pimatic aplinką, tai kuriamas projektas turi atspindėti šių duomenų apsikeitimą ir atvaizdavimą grafinėje sąsajoje. Tiek Vhduino.c kodas, tiek Pimatic aplinkos failai yra įdiegti virtualioje mašinoje, kuri pastatyta ant Linux operacinės aplinkos. Čia Vhduino.c kodas ir Pimatic bendrauja dviem tarpusavyje susietais įrenginiais (angl. *serial port pair*) `tnt0` ir `tnt1`. Jų veikimos principas yra toks, kad įrašytų duomenis iškart siunčia į susietą įrenginį ir taip nuolatos keičiasi duomenimis.

Norint atvaizduoti tokį duomenų apsikeitimą grafinėje sąsajoje tenka pagalvoti apie taip, kaip toks duomenų apsikeitimas bus įgyvendimas, kai Pimatic aplinka yra įdiegta Linux virtualioje mašinoje, o grafinė sąsaja paleidžiama asmeniniame Windows kompiuteryje. Tam bus naudojama technologija `ser2net`.

1.2. PROJEKTO FUNKCIONALUMAS

Šis projektas didelio funkcionalumo neturi. Pagrindinė šio projekto priežastis yra pagerinti vartotojo supratimą apie sistemos veikimą nereikalaujant papildomų išlaidų (t.y. nereikia turėti fizinio ArduinoNano modelio, DHT sensoriaus ar pnš.).

Vienintelis projekto funkcionalumas yra greitas ir nesudėtingas virtualios mašinos ir personalinio kompiuterinio sujungimas internetu naudojant `ser2net` technologiją, kai personaliniame kompiuteryje yra paleista ArduinoNano grafinė sąsaja, o virtualioje mašinoje Pimatic servisas ir Vhduino.c kodas.

1.3. NAUDOTI ĮRANKIAI PROJEKTO KŪRIMUI

Pagrindiniai įrankiai naudoti projekto kūrimui yra `ser2net` technologija ir Processing4.3. `ser2net` technologija leidžia TCP protokolu bendrauti dviem IP adresams, kurie gali ir nepriklausyti tam pačiam tinklui. Šiuo atveju jungsime asmeninį Windows kompiuterį su virtualia mašina (Linux). Virtualių port'ų kūrimui naudojama VSPE programinė įranga. Grafinei sąsajai atvaizduoti ir duomenims siųsti naudojamas Processing4.3, kuris remiasi Java programavimo kalb.

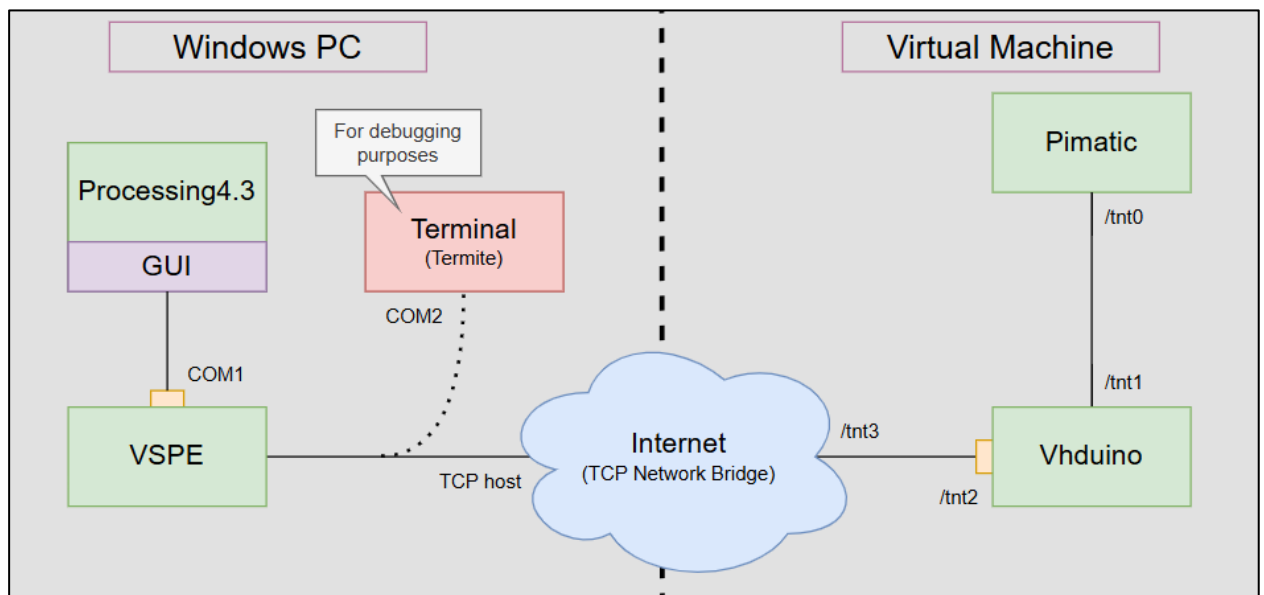
1.4.KŪRIMO PROCESAS

Kūrimo procesas susideda iš kelių pagrindinių dalių:

1. ser2net instaliacija, jos konfigūracinių failų pakeitimas ir serviso paleidimas;
2. VSPE konfigūravimas;
3. Vhduino.c kodo papildymas, kad duomenys būtų siunčiami ir į kitą virtualių port'ų porą;
4. Processing4.3 kodo paleidimas.

1.5.PROJEKTO SCHEMA

Žemiau pateikiama realizuoto projekto schema:



1 pav. Projekto schema

1.6. IŠŠŪKIAI IR IDĖJOS ATEIČIAI

Norint išgauti papildomą funkcionalumą reiktų pagalvoti apie tokius galimus patobulinimus:

1. ArduinoNano modulį pakeisti bet koku kitu modeliu arba sukurti savo asmeninį, kad ir su 1000 pin'ų;
2. Įdiegti MQTT brokerį, kad Pimatic būtų galima valdyti iš kito įrenginio nuotoliu;
3. Papildyti grafinę sąsają naujais komponentais/technologiniais sprendimais;
4. Užbaigti projekto idėją sutvarkant Processing4.3 mygtukų funkcionalumą, kad spaudant mygtukas grafinėje sąsajoje, jų paspaudimai atsivaizduotų Pimatic aplinkoje.

2. KAIP NAUDOTIS PROJEKTU?

1. Įdiekite ser2net:

```
sudo apt install ser2net -y
```

2. Patikrinti ar diegimas pavyko:

```
ser2net --version
```

3. Atnaujinkite savo ser2net.conf ir ser2net.yaml failus aplanke /etc (arba tiesiog atsisiųskite juos iš github saugyklos ir pakeiskite senus failus):

```
sudo nano /etc/ser2net.conf  
sudo nano /etc/ser2net.yaml
```

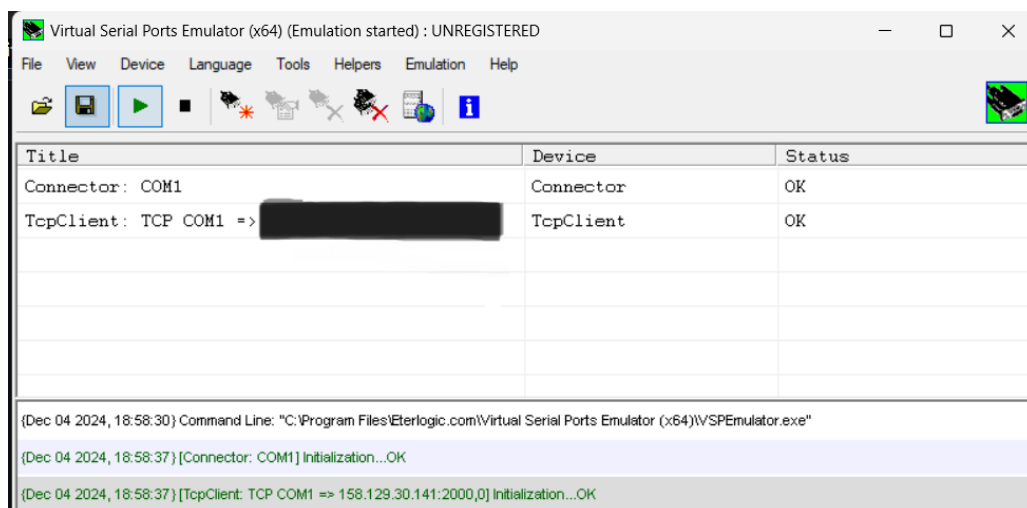
4. Įjungti / paleisti iš naujo / sustabdyti arba patikrinti ser2net paslaugos būseną (nepamirškite perkrauti ser2net, jei pakeitėte .yaml ar .conf failus):

```
sudo systemctl enable ser2net  
sudo systemctl restart ser2net  
sudo systemctl start ser2net  
sudo systemctl status ser2net
```

5. Paleidus komandą “sudo systemctl status ser2net” turėtūsi matyti kažkas panašaus į tai:

```
root@t120b139-11:~# sudo systemctl status ser2net  
● ser2net.service - Serial port to network proxy  
   Loaded: loaded (/lib/systemd/system/ser2net.service; enabled; vendor preset: enabled)  
   Active: active (running) since Mon 2024-12-02 13:12:39 UTC; 1 month 12 days ago  
     Docs: man:ser2net(8)  
   Main PID: 283906 (ser2net)  
     Tasks: 1 (limit: 2219)  
    Memory: 1.7M  
       CPU: 23min 39.323s  
    CGroup: /system.slice/ser2net.service  
            └─283906 /usr/sbin/ser2net -n -c /etc/ser2net.yaml -P /run/ser2net.pid  
  
Dec 02 13:12:39 t120b139-11 systemd[1]: Starting Serial port to network proxy...  
Dec 02 13:12:39 t120b139-11 systemd[1]: Started Serial port to network proxy.
```

6. Atidarykite VSPE ir įkelkite konfigūraciją. Pakeiskite IP adresą TcpClient įrenginyje į tokį patį, kaip jūsų virtualios mašinos adresas. Prievadą palikite tokį patį. VSPE aplinka turėtų atrodyti maždaug taip:



7. Kai ser2net veikia ir VSPE sukonfigūruotas, laikas paleisti Processing 4.3 kodą. Jei jis paleidžiamas, tai reiškia, kad prisijungta prie tinklo.
8. Perkopijuokite VhduinoU.c kodą iš savo kompiuterio į virtualią mašiną su komanda:

```
scp C:\Users\[YourWindowsUsername]\Desktop\Patobulinta\VhduinoU.c  
root@158.129.30.xxx:/root
```

9. Sukompiliuokite kodą virtualioje mašinoje su:

```
gcc -o VhduinoU VhduinoU.c
```

10. Viskas baigta. Paleiskite pimatic aplinką su „pimatic.js start“ viename virtualios mašinos lange, o kitame lange paleiskite VhduinoU kodą su „/VhduinoU“.

3. GITHUB

GitHub nuoroda į projektą: <https://github.com/Kristupelis/ArduinoNano-GUI-for-Pimatic>.

IŠVADOS

Darbas pavyko ir buvo labai įdomus. Komunikacija tarp Pimatic ir Processing4.3 veikia puikiai, ryšys spartus ir nenutrūkstamas. Projektas nėra galutinai užbaigtas ir jam galima suteikti daugiau papildomų funkcionalumų.