

Přenos telemetrie přes IoT



Bc. Filip Malý
Vedoucí: Ing. Tomáš Vokoun

Cíl práce

Hlavní cíl:

- ❑ Přenos telemetrie prostřednictvím Internet of Things (IoT) rozhraní

Dílčí cíle:

- ❑ Analýza IoT rozhraní včetně porovnání se stávajícími nejběžnějšími poskytovateli IoT
- ❑ Tvorba aplikace pro komunikaci s telemetrickým zařízením prostřednictvím IoT
- ❑ Implementace komunikace s prototypem dronu

Metodika

Teoretická část:

- ❑ Rešeršní část analýza odborné a vědecké literatury

Praktická část:

- ❑ Metody vícekriteriální analýzy variant (Bodovací a Saatyho metoda)
- ❑ Charakteristika telemetrie pomocí UML
- ❑ Aplikace vytvořena v prostředí SAP FIORI
- ❑ Jazyky:
 - ❑ JavaScript
 - ❑ Python
 - ❑ SQL
- ❑ Shell scripting

Kritéria pro vícekriteriální analýzu variant

- ▶ 16 kritérií
- ▶ Váhy Saatyho metodou
- ▶ 10 nejvýznamnějších kritérií

Pořadí	Kritérium	Váha
1	Cena	0,180421
2	Licence	0,147655
3	Propojení na Github	0,108668
4	Databáze	0,10638
5	Konektivita	0,075792
6	Hardware podpora	0,06966
7	Protokoly	0,058828
8	Vlastní knihovny	0,057941
9	API dokumentace	0,043526
10	Vlastní sada aplikací	0,035387

Výsledky analýzy IoT

- ▶ Zaměřeno na přenos telemetrie z dronu
- ▶ Srovnání 20 poskytovatelů
- ▶ Bodovací metoda

5 nejlépe hodnocených	Poskytovatel	Body
1	SAP IOT LEONARDO	1405
2	IBM Watson IoT Platform	1310
3	Microsoft Azure IoT hub	1225
4	Google Cloud IoT	1195
5	Samsung ARTIK Cloud	1190

- ▶ Bodovací metoda ohodnocená o váhy ze Saatyho metody

5 nejlépe hodnocených	Poskytovatel	Body
1	SAP IOT LEONARDO	87,17527521
2	IBM Watson IoT Platform	79,10420544
3	Google Cloud IoT	77,41640078
4	GE Predix	74,11819792
5	Samsung ARTIK Cloud	74,03684658

Aplikace pro komunikaci s telemetrickým zařízením

Základní info + IoT správa dronu (Bc. Filip Malý 2018 ©) Refresh neaktivní Vybraný dron: CZU_HEXA

dbRefresh 3 tabulka O aplikaci

em. délka Aktuální pozice

Výběr dronu

CZU_HEXA

CZU_HEXA

CZU_QUAD

Aneta Koudelkova

Poslední použití

6.10.2017

Délka posledního letu

5.10.2017

Čas poslední cesty

15

Počet nálétaných KM

125

Map showing flight path of drone CZU_HEXA over Prague area, including locations like Staré Dáblice, Praha-Dáblice, Staré Letňany, Praha 18, Letov, and Letiště Praha-Kbely.

Monitoring Requesty Fotografie DB tabulka O aplikaci

Záznamy z letu

<input type="checkbox"/>	Datum	Model	Napětí
<input type="checkbox"/>	23. 1. 2018 22:20:20	3	121
<input type="checkbox"/>	23. 1. 2018 22:20:08	3	121
<input type="checkbox"/>	23. 1. 2018 22:19:59	3	121

Vytvořené prototypy

- ▶ Quad dron s RPI 2



- ▶ Hexa dron s RPI 3



Přenesená telemetrie

▶ Příkaz z aplikace

The screenshot shows the 'Request Headers' and 'Request Payload' sections of a web browser's developer tools. The request is a POST to a URL with a long ID. The headers include 'Content-Type: application/json; charset=UTF-8' and 'Accept: */*'. The payload is a JSON object with a 'messageType' and a 'messages' array containing a single message object with 'timestamp', 'deviceId', and 'commandID' fields.

```
Request URL: https://webidetesting4826469-s0015766703trial.dispatcher.hanatrial.ondemand.com/iotms/push/46b877a1-1639-4e79-b86d-1eb8a5e92421
Request Method: POST
Status Code: 202 Accepted
Remote Address: 155.56.219.29:443
Referrer Policy: no-referrer-when-downgrade

Response Headers
Content-Length: 76
Content-Type: application/json; charset=UTF-8
Date: Fri, 23 Mar 2018 22:05:05 GMT
Server: SAP
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload

Request Headers
Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: cs,en;q=0.9,en-GB;q=0.8
Cache-Control: no-cache
Connection: keep-alive
Content-Length: 436
Content-Type: application/json
Cookie: JTENANTSESSIONID_7b1b1837-b36b-42de-b045-c9c3780d9766-2V6W0ZAZH2K2F82vts5qTJKJqSG6w2Fa8q708o1s1QzD0nK30; JTENANTSESSIONID_s0015766703trial=iJXk2F10rZhAPtArLo118Vf8pCAl7b2kPm6N2b0tntVtIE
%#3D; BIGipServer-jpaas_folder-dispatcher.hanatrial.ondemand.com~1x0~CsSDQ/In9dFGwOhtCRHmTAT63a50p1u561y4ldqnsFXJEdXje9JQ7K1HFk6k8Z751RN12Q2A***; oucrsoheronkhyctnbngr1yp=KwJ30Yea1d91dyP%2b25F
VxpeEeGMkjd7WUUC4j305b1Zms29e2FSt0P8b07L8s6Usho0h134jvS2611n1UusnT12K2F2bCCq%2FtLExj4jV0Upze1rWAl10i08F1wBEdqXmYl0yMjcmjxnP1aFw0Q63JST289C~LcknB8ro5XoZehurPLFbplgKuLxj4jLothB6E67aK2FV
cHLSV8QK2FvW28s2F5KXk28p1cukchq1XovXVF7UdeZnoU1hIdfM0Uj2zFVFD0uMCP1mnsVjEFFHsvj8j0s205FDtqAPhge1eKRkUSGAI3UNk28Vb05ccvZNCxkLkukDMvRFR71AIs7j0noVsRk6RvF6UAdk1yHyV66zChU9UjYkHingLbHqB7eAVbq
wqkeuvgUq%3DnK30; JSESSIONID=6057C509A72082C284827037F5F802323C14850EFC06862698E0E59E185A5668
Host: webidetesting4826469-s0015766703trial.dispatcher.hanatrial.ondemand.com
Origin: https://webidetesting4826469-s0015766703trial.dispatcher.hanatrial.ondemand.com
Pragma: no-cache
Referer: https://webidetesting4826469-s0015766703trial.dispatcher.hanatrial.ondemand.com/webapp/extended_runnable_file.html?h1c_onionpath=K2F8015766703trialN2450015766703-OrionContent%2FDRONE_IOT8o
%Regional-url-index.html&sp-u1-espCacheBuster=.X2F8hc_destmappng.1otms-8sasp-u1-xx-componentPreLoad=off
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.186 Safari/537.36
X-Requested-With: XMLHttpRequest

Request Payload
{
  "method": "http",
  "messageType": "1a6fa20042f203191551",
  "sender": "IOT APP",
  "messages": [
    {
      "timestamp": "2018-03-23T22:05:05.597Z",
      "deviceId": 123,
      "commandID": 1,
      "method": "http",
      "sender": "IOT APP"
    }
  ]
}
```

▶ Telemetrie v dronu

The screenshot shows a Python IDE with a project named 'IOT_TELEMETRY'. The code in 'Main.py' defines a 'Telemetry' class with methods for sending telemetry data and handling requests. The 'startProgram' method calls 'sendTelemetry' and 'getTelemetryData'. The 'getTelemetryData' method returns a list of messages. The 'requests.get' method is used to handle incoming requests. The 'sendTelemetry' method sends a POST request to a server. The 'getTelemetryData' method returns a list of messages. The 'startProgram' method calls 'sendTelemetry' and 'getTelemetryData'. The 'getTelemetryData' method returns a list of messages. The 'requests.get' method is used to handle incoming requests. The 'sendTelemetry' method sends a POST request to a server.

```
self.camera.flip = True
self.camera.flip = True
except Exception:
    print "Camera not initialized!" + str(error)

def startProgram(self):
    counter = 1
    try:
        while True:
            self.sendTelemetry(self.getTelemetryData(), self.config.getDeviceId(), self.config.getTelemetryReader())
            self.getResponse()
            time.sleep(self.config.getFromMetadata('sleep'))
            print counter
            counter = counter + 1
        except Exception:
            print "Ochop v cyklu programu" + str(error)

def getResponse(self):
    self Telemetry = self Telemetry instance at 0x71d4f170
    response = requests.get("https://" + self.config.getHost() + self.config.getDeviceId(), headers=self.config.getRequestHeader())
    print "Task completed" + str(response.status_code)

if response.status_code == 200:
    data = json.loads(response.text)
    print "Response: %s" % data
    print "Data: %s" % data

    Telemetry.getResponse() if response.data() for x in range(10)
```


Závěr práce

- ▶ Teoretická část
 - ❑ Popis základní problematiky
 - ❑ Odborné a vědecké zdroje v elektronické podobě
- ▶ Praktická část
 - ❑ Vybrání nejvhodnějšího poskytovatele vícekriteriální analýzou variant
 - ❑ Responsivní HTML aplikace vytvořená jazyky XML a JavaScript
 - ❑ IoT modul pro komunikaci se SCP v jazyce python
 - ❑ Dva prototypy dronu s RPI 2 a 3
 - ❑ Nastavení IoT služby
 - ❑ Nastavení databáze a vytvoření xsodata služeb
 - ❑ Obousměrné přenesení telemetrie