

Projekt EVC

Co přesně každá z nás udělala

Daria Roshchupkina

Napsala jsem všechny kód v `src/BaliseGroup.hpp`, `src/CallbackObjectsEVC.hpp`, `src/CallbackObjectsEVC.cpp`, `src/EVC.cpp`, `src/MessageInfo.hpp`, `src/ReportHelper.hpp`. Udělala jsem spolu s Alinou `src/Client.hpp`. V `src/Client.cpp` jsem napsala všechno v konstruktoru a v metodách `ReceiveDMI`, `ReceiveBTM`, `SendRBC`, `SendDMI`, `SendTIU`, `DMIToJson`, `MakeReport`, `SendPositionReport`, `SendTerminationCommunication`, `CheckTimestamp`. Také jsem napsala všechno v `ReceiveRBC` kromě kódu v case 3. Napsala jsem vytvoření a posílání zpráv RBC v metodách `ReceiveODO` a `ReceiveTIU`. Ke svému kódu v projektu EVC jsem napsala komentáře. Udělala jsem `CMakeLists.txt` a `.gitignore`. V průběhu implementace jsem testovala metody třídy `Client`.

Alina Krasnenkova

Napsala jsem všechny kód v `src/Gradient.hpp`, `src/Gradient.cpp`, `src/GradientProfileEVC.hpp`, `src/GradientProfileEVC.cpp`, `src/ODO.hpp`, `src/SSP.hpp`, `src/SSP.cpp`, `src/SpeedProfileEVC.hpp`, `src/SpeedProfileEVC.cpp`, `src/TIU.hpp`. Udělala jsem spolu s Dariou `src/Client.hpp`. V `src/Client.cpp` jsem napsala všechno v metodách `TIUToJson`, `BrakingCurve`, `BrakingCurveNew`, `ChangeStatus`, `ChangeSection`, `Scale`, `CutGradient`, `CutSSP`. V metodách `ReceiveODO` a `ReceiveTIU` jsem udělala všechno kromě vytvoření a posílání zpráv RBC. V `ReceiveRBC` jsem udělala jenom case 3. Ke svému kódu v projektu EVC jsem napsala komentáře. V průběhu implementace jsem testovala metody třídy `Client`.

Popis projektu EVC

Projekt EVC je napsán pro demo verzi simulátoru ETCS, avšak v budoucnu ho lze jednoduše rozšířit na další verze simulátoru ETCS. Aplikace EVC je jednovláknová.

Projekt se nachází na gitlabu: <https://gitlab.fit.cvut.cz/etcs/evc.git>. Projekt používá src/MQTTClient, src/MessagesAndPackets a src/BrakingCurve jako subtrees.

MessagesAndPackets se nachází na gitlabu:

<https://gitlab.fit.cvut.cz/etcs/messagesandpackets.git>. MQTTClient se nachází na gitlabu:

<https://gitlab.fit.cvut.cz/etcs/mqttclient.git>. BrakingCurve se nachází na gitlabu:

<https://gitlab.fit.cvut.cz/bilekm10/BrakingCurve.git>.

Projekt se řídí informacemi z SUBSET-026-3 ISSUE : 2.3.0, SUBSET-026-4 ISSUE : 2.3.0, SUBSET-026-5 ISSUE : 2.3.0, SUBSET-026-7 ISSUE : 2.3.0, SUBSET-026-8 ISSUE : 2.3.0 a ERTMS UNIT ETCS DRIVER MACHINE INTERFACE Version: 2.3.

Celý projekt je dobře okomentován. V komentářích je používána zkratka „due to subsets' standards“, která odkazuje na subsety uvedené výš.

V ETCS projektu pro demo verzi existuje 6 různých modulů: EVC, RBC, DMI, TIU, ODO, BTM, které mezi sebou komunikují. TIU, ODO, BTM, DMI posílají data EVC. EVC posílá data jen TIU a DMI. EVC posílá zprávy RBC. RBC posílá zprávy EVC.

Hlavním úkolem EVC je přijímat zprávy od modulu RBC a data od modulů DMI, TIU, ODO a BTM a na základě přijatých informací provádět nějaký konkrétní děj, případně odesílat zprávy či data modulům. Odesílání zpráv a dat modulům se provádí jenom v metodách přijetí informací od modulů, t.j. EVC nemůže poslat modulu informace bez přijetí informací od modulu. Například EVC přijme informace od ODO a zpracuje je v metodě ReceiveODO, kde dojde ke změně modu, a tedy i k zaslání zprávy Train Position Report RBC.

Tým EVC se domluvil s týmem RBC, že v prvním MA, který posílá RBC EVC musí být packet 58 (Position Report Parameters). Na základě hodnot v proměnných D_CYCLOC a Q_SCALE z tohoto packetu EVC posílá Train Position Report RBC. Pro toto posílání není vytvořené samostatné vlákno z důvodu, že data z modulu ODO se odesílají každých 10 ms, tedy velmi často, a proto kontrola splnění požadavku, že vlak projel už dalších několik metrů, probíhá během zpracování informací přijatých z ODO.

Komunikace s ostatními moduly

Komunikace s ostatními moduly probíhá v JSON pomocí MQTT brokeru.

Komunikace EVC <-> RBC

Pro komunikaci s RBC EVC používá zprávy a packety z src/MessagesAndPackets.

Pořadí odeslání zpráv mezi EVC a RBC v demo verzi. Po odeslání zprávy MA Request, EVC může přijímat jen Movement Authority od RBC a odesílat RBC jen Train Position Report do okamžiku, než odešle RBC End of Mission.

Messages from Train

1

Initiation_of_a_communication_session
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint32_t NID_ENGINE

3.1

Session_established
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint32_t NID_ENGINE vector<Packet> packets

3.2

No_compatible_version
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint32_t NID_ENGINE

3.2.1

Termination_of_a_communication_session
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint32_t NID_ENGINE

3.1.1

SoM_Position_Report
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint32_t NID_ENGINE uint8_t Q_STATUS vector<Packet> packets

Messages from RBC

2

Configuration_determination
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint8_t M_ACK uint32_t NID_LRBG uint8_t M_VERSION

3.2.1.1

Acknowledgement_of_termination_of_a_communication_session
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint8_t M_ACK uint32_t NID_LRBG

3.1.2

Validated_train_data
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint32_t NID_ENGINE vector<Packet> packets

4

Acknowledgement_of_Train_Data
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN0 uint8_t M_ACK uint32_t NID_LRBG uint32_t T_TRAIN1

5

MA_Request
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint32_t NID_ENGINE bool Q_TRACKDEL vector<Packet> packets

6

Movement_Authority
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint8_t M_ACK uint32_t NID_LRBG vector<Packet> packets

7

Train_Position_Report
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint32_t NID_ENGINE vector<Packet> packets

8

End_of_Mission
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint32_t NID_ENGINE vector<Packet> packets

9

Termination_of_a_communication_session
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint32_t NID_ENGINE

10

Acknowledgement_of_termination_of_a_communication_session
uint8_t NID_MESSAGE uint16_t L_MESSAGE uint32_t T_TRAIN uint8_t M_ACK uint32_t NID_LRBG

Komunikace EVC <- BTM

Struktura JSON objektu, který se odesílá z modulu BTM EVC. Informace jsou převzaté z 8.4.2 Rules for Eurobalise telegrams z SUBSET-026-8 ISSUE : 2.3.0 a z 7.5 Definitions of Variables a 7.4.4PACKETS: TRACK TO TRAIN or TRAIN TO TRACK 7.4.4.1 Packet Number 255: End of Information z SUBSET-026-7 ISSUE: 2.3.0.

Q_UPDOWN

- Means: The direction of the information: Down-link telegram (train to track) (0) , Up-link telegram (track to train) (1).
- Type: number.
- **Q_UPDOWN always = 1.**

M_VERSION

- Means: Version of the ERTMS/ETCS language.
- Type: number.
- **M_VERSION always = 16.**

Q_MEDIA

- Means: The type of media: Balise (0).
- Type: number.
- **Q_MEDIA always = 0.**

N_PIG

- Means: Position in the group. Defines the position of the balise in the balise group (0 - I am the first, 1 - I am the second ...).
- Type: number.
- Values: from 0 to 7.

N_TOTAL

- Means: Total number of balises in the balise group (0 - 1 balise in the group, 1 - 2 balise in the group ...).
- Type: number.
- **N_TOTAL always = 1.**

M_DUP

- Means: Indication whether the information of the balise is a duplicate of the balise before or after this one.
- Type: number.
- **M_DUP always = 0.**

M_MCOUNT

- Means: Telegram counter (M_MCOUNT) - 8 bits. To enable detection of a change of balise group message during passage of the balise group.

- Type: number.
- **M_MCOUNT always = 0.**

NID_C

- Means: Country or region.
- Type: number.
- Values: from 0 to 1023.

NID_BG

- Means: Identity of the balise group.
- Type: number.
- Values: from 0 16382. (1638 – can't be, cause there is no linking in demo).

Q_LINK

- Means: Marks the balise group as linked (Q_LINK = 1) or unlinked (Q_LINK = 0).
- Type: number.
- **Q_LINK always = 0.**

End of Information – packet number 255

- Means: End of information.
- Type: object.
- Includes:
 - **NID_PACKET**
 - Means: packet identifier, which is used in **End of Information**.
 - Type: number.
 - **NID_PACKET always = 255.**

Příklad JSON objektu, který se odesílá z BTM EVC.

```

1  {
2    "Q_UPDOWN" : 1,
3    "M_VERSION" : 16,
4    "Q_MEDIA" : 0,
5    "N_PIG" : 0,
6    "N_TOTAL" : 1,
7    "M_DUP" : 0,
8    "M_MCOUNT" : 0,
9    "NID_C" : 12,
10   "NID_BG" : 14,
11   "Q_LINK" : 0,
12   "End of Information" : {
13     "NID_PACKET" : 255
14   }
15 }

```

Komunikace EVC < - ODO

EVC přijímá aktuální rychlost vláku z modulu ODO.

Struktura JSON objektu, který se odesílá z modulu ODO EVC.

train_speed

- Means: Current train speed [m/s].
- Type: number

Příklady JSON objektu, který se odesílá z ODO EVC.

```

1  {
2    "train_speed" : 100
3  }

```

```

1  {
2      "train_speed" : 100.5
3  }

```

Komunikace EVC < - > TIU

EVC přijímá informace o aktivaci stanoviště a stavu jízdni páky od TIU.

Struktura JSON objektu, který se odesílá z modulu TIU EVC.

battery_power

- Means: Battery power.
- Type: boolean.
- Values: 1 – power is on, 0 – otherwise.

cab

- Means: Cab is active.
- Type: boolean.
- Values: 1 – cab is active, 0 - otherwise

train_direction

- Means: Direction if train movement in relation to the balise group.
- Type: number.
- Values: 0 – reverse, 1 – nominal, 2 – unknown

Příklady JSON objektu, který se odesílá z TIU EVC.

```

1  {
2      "battery_power" : true,
3      "cab" : true,
4      "train_direction" : 1
5  }

```

EVC posílá na TIU informace o aktivaci průběžné nebo nouzové brzdy.

Struktura JSON objektu, který se odesílá TIU z EVC.

service_brake

- Means: Service brake must be triggered.
- Type: boolean.
- Values: true – service brake must be triggered, false – otherwise.

emergency_brake

- Means: Emergency brake must be triggered.
- Type: boolean.
- Values: true – emergency brake must be triggered, false- otherwise.

Příklady JSON objektu, který se odesílá TIU z EVC.

```
1  {  
2      "emergency_brake" : true,  
3      "service_brake" : false  
4  }
```

Komunikace EVC < - > DMI

Struktura JSON objektu, který se odesílá z modulu DMI EVC.

driver_id

- Means: Driver entered id.
- Type: boolean.
- Values: true - Driver entered valid id, false - driver didn't enter id or it's invalid.

start

- Means: Start of mission.
- Type: boolean.
- Values: true - Driver pressed Start on display, false – otherwise.

Příklady JSON objektu, který se odesílá z DMI EVC.

```
1  {  
2      "start" : false,  
3      "driver_id" : false  
4  }
```

Struktura JSON objektu, který se odesílá z EVC DMI.

session_established

- Means: Session with RBC was established.
- Type: boolean.
- Values: true - Session with RBC was established, false - otherwise.

cabin

- Means: Cab is active.
- Type: boolean.

- Values: true - cab is active, false - otherwise.

ma

- Means: First MA was received from RBC.
- Type: boolean.
- Values: true - First MA was received from RBC, false - otherwise.

mode

- Means: Mode.
- Type: number.
- Values: 0 - Full Supervision, 6 - Stand By, 7 - Trip, 15 - No Power.

train_speed

- Means: Current train speed, km/h.
- Type: number.
- Values: from 0 to max train speed.

permitted_speed

- Means: Permitted train speed, km/h.
- Type: number.
- Values: from 0 to max train speed.

intervention_speed

- Means: FLOI train speed, km/h.
- Type: number.
- Values: from 0 to max train speed.

target_speed

- Means: Target train speed, km/h.
- Type: number.
- Values: from 0 to max train speed.

target_distance

- Means: Target distance, m.
- Type: number.
- Values: from 0 to 5000 (due to track distance for demo version).

GradientProfile

- Means: Structure for keeping Gradient Profile received by RBC.
- Type: object.
- Includes:
 - Q_SCALE
 - Means: Qualifier for the distance scale.
 - Type: number.
 - Values: 0 - 10 cm scale, 1 - 1 m scale, 2 - 10 m scale.
 - m_Gradients
 - Means: Array of gradients. The first value in array corresponds to the front end of the train.
 - Type: array.
 - Includes objects, every of them consists of:
 - ❖ D_GRADIENT
 - Means: Incremental distance to next change of gradient, cm or m depending on Q_SCALE.
 - Type: number.

- Values: from 0 to 32767.
- ❖ Q_GDIR
 - Means: Qualifier for gradient slope.
 - Type: number.
 - Values: 0 - downhill, 1 - uphill.
- ❖ G_A
 - Means: Safe gradient [%].
 - Type: number.
 - Values: from 0 to 254.

SpeedProfile

- Means: Structure for keeping Speed Profile received by RBC.
- Type: object.
- Includes:
 - Q_SCALE
 - Means: Qualifier for the distance scale.
 - Type: number.
 - Values: 0 - 10 cm scale, 1 - 1 m scale, 2 - 10 m scale.
 - m_SSps
 - Means: Array of speed profiles.
 - Type: array.
 - Includes objects, every of them consists of:
 - ❖ D_STATIC
 - Means: Incremental distance to next discontinuity in a international SSP profile.
 - Type: number.
 - Values: from 0 to 327670.
 - ❖ V_STATIC
 - Means: Static speed profile.
 - Type: number.
 - Values: from 0 to 600.

supervision_section

- Means: Monitoring section.
- Type: number.
- Values: 0 - CSM, 1 - PIM, 2 - TSM.

supervision_status

- Means: Supervision status.
- Type: number.
- Values: 0 - NoS, 1 - OvS, 2 - WaS, 3 - IntS, 4 - IndS (only in TSM).

pre_indication_location

- Means: Pre-indication location, m.
- Type: number.
- Values: from 0 to 5000 (due to track distance for demo version).