Brain-robot communications in the Internet of Things

SVETOSLAV EMILOV KREMENSKI AND ANNA KOSTADINOVA LEKOVA

INSTITUTE OF ROBOTICS, BULGARIAN ACADEMY OF SCIENCES, SOFIA, BULGARIA

Outline

- Introduction
- Used tools and APIs
- Methods and algorithms for BCI - Robot communication in IoT
- EPOC – Pepper communications via Node-RED
- Conclusion

Introduction

We propose a novel approach for integrating EEG brain computer interface (BCI) with a humanoid robot for communications in the IoT.

- different BCI technologies and humanoid robots to be connected as IoT based devices and services using Node-RED as an integrated platform
- how a BCI device and a robot are transformed into IoT devices (*things*) using cross-platform server side runtime environment built in browser and how these *things* share and collect data with minimal human intervention based on publish-subscribe messaging pattern
- we validated the proposed approach by integrating the Emotiv EPOC+ EEG headset and the humanoid robot Pepper in Node-RED

The reported in the scientific community solutions for EEG-based Brain-Robot communications in the IoT are platform, device and application specific.

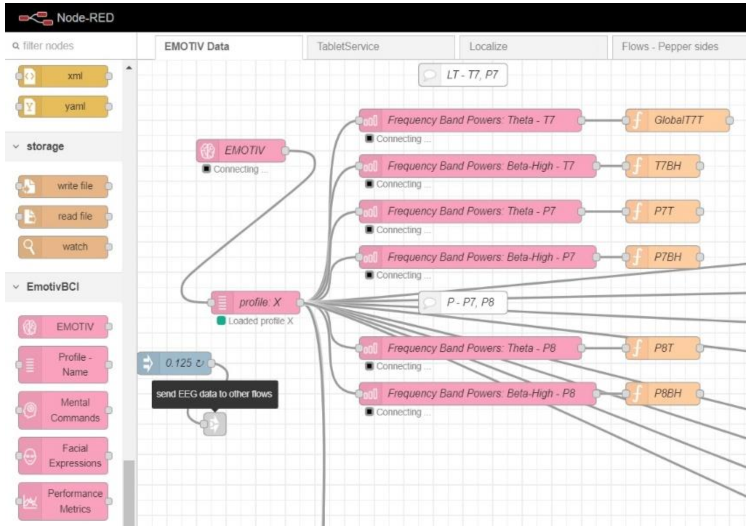
USED TOOLS AND APIs

- ❑ Node-RED
  - browser-based flow programing tool based on Node.js and provide a gateway to IoT
  - via the Emotiv-BCI nodes in Node-RED the EPOC+ device interface *things* to share and collect data

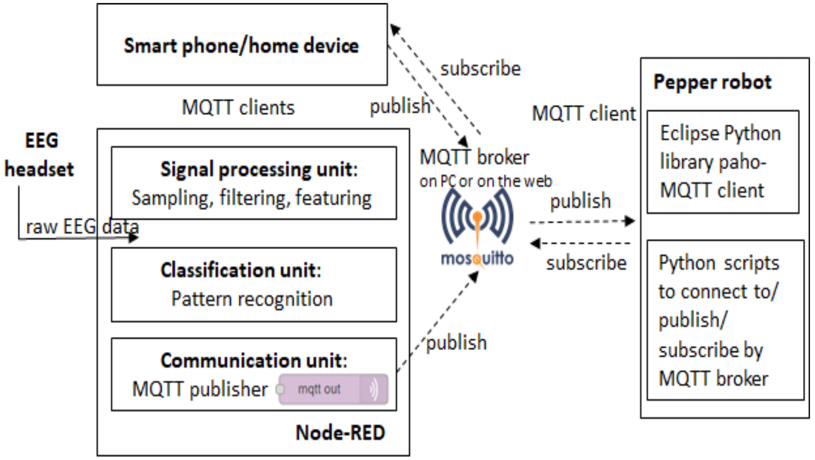
- ❑ Robot Pepper
  - different modalities for interaction, such as gestures, speech, audio playing, words recognition, and executes a planar move. Also has people-perception module and an Android based tablet on its chests
  - can be used for active interaction or to show presentations, videos, animations, web content, metrical maps, as a social agent into retail, banking, social care, hotels and restaurants
  - has a preinstalled Operating System - NAOqi or Android, we use *NAOqi*. This framework supports communications between different modules, as well as the programming and information sharing
  - Information for the actual state of sensors, actuators and events in terms of key-value variables are recorded in the internal read-write memory of the robot and is accessible via *NAOqi* Application Programming Interface (API)

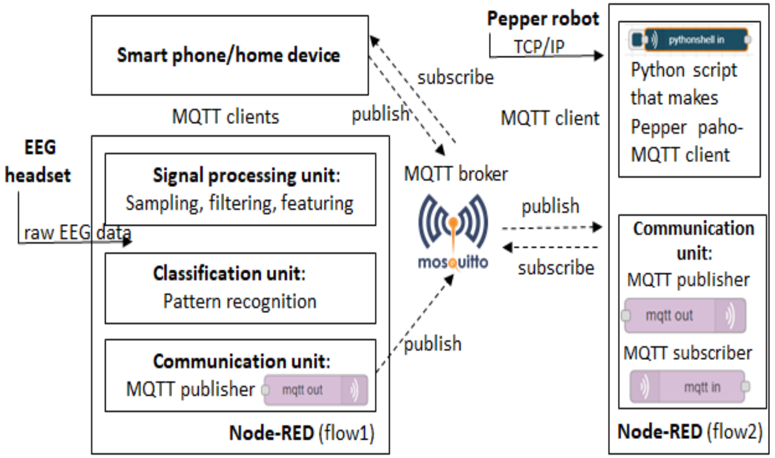
Methods and algorithms for BCI – Robot communication in IoT

- We designed and developed flow in Node-RED to register, record and send EEG data to the other flows in Node-RED, one of which is the flow that interfaces the Pepper APIs
  - we applied the Emotiv library for Node-RED that provides nodes for gaining the theta and high beta frequencies band powers for the electrodes F3, F4, FC5, FC6, T7, P7, P8, O1 and O2 of the BCI device EPOC+
  - we used EPOC+ with a sampling resolution of 256 Hz, resulting in 256 row EEG samples per second and one band power for each 32 samples. These data were sent to the other Node-RED flows every 125 millisecond via a *link-out node*

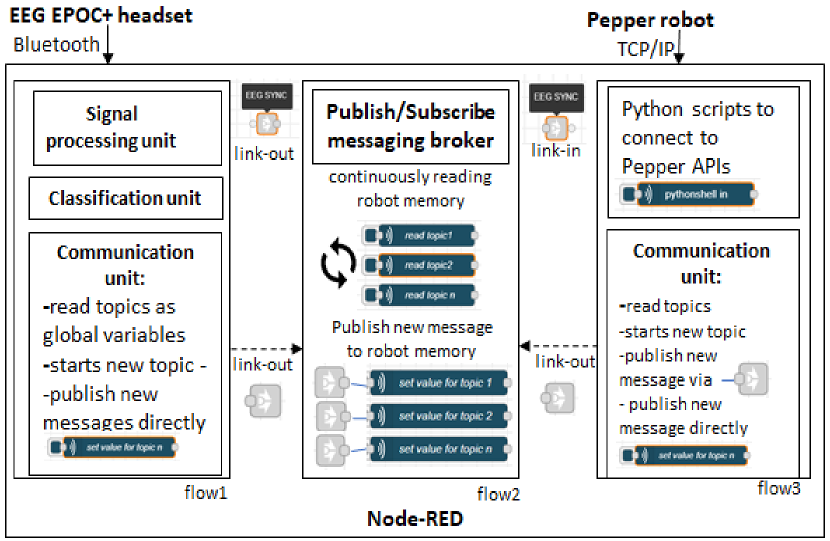


- ❑ Transforming Pepper into IoT device by applying the MQTT protocol on the robot side in order to connect it to MQTT broker, which listens to data published to topics
  - switched to qi Framework that initializes and get qi.Session from NAOqi
  - used the open source Eclipse Mosquitto broker running locally on a PC
  - a Python library paho-MQTT for Eclipse mosquito was imported from a console application that initialized the qi framework and connected to the main session of the robot
- ❑ The second tested approach for transforming Pepper into IoT device also used the Eclipse Mosquitto broker locally or on the web, however the connection to the broker and the messages exchange were established by *mqtt-in* and *mqtt-out* nodes in Node-RED
  - The robot as an MQTT client connected to the broker by calling the NAOqi APIs from Node-RED instead from external IDE
  - Thus, Pepper used *mqtt-in* and *mqtt-out* to publish-subscribe in Node-RED
  - After publishing a message “wake up“ by a *mqtt-out* node to the mosquito broker, the message is received as a payload of a *mqtt-in* node via the Python paho-MQTT client on the robot
  - Based on the received message payload and topic, the parameters for the python script are defined and sent to the *pythoshell node* for calling *NAOqi* APIs





- Novel way for transforming Pepper into IoT device by only Node-RED, which follows the event-driven architecture realized through publish-subscribe communication model and allows devices to interact asynchronously
  - We used the robot memory as a data broker and read/write in it
  - Via a continuous python process in a Node-RED flow we call to *ALMemory APIs* in order to provide the current value for the keys (topics) of interest



EEG EPOC+ headset

Bluetooth

Pepper robot

TCP/IP

Signal processing unit

Classification unit

Communication unit:
-read topics as global variables
-starts new topic -
-publish new messages directly

Publish/Subscribe messaging broker

continuously reading robot memory

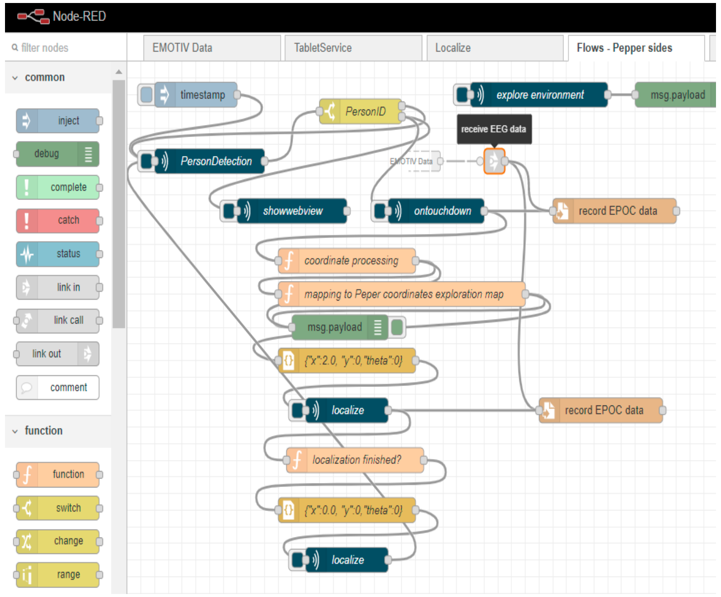
Publish new message to robot memory

Python scripts to connect to Pepper APIs

Communication unit:
-read topics
-starts new topic
-publish new message via
- publish new message directly

❑ *Design of the flow for the humanoid robot*

- developed is flow for Pepper communication with the messaging broker, EEG data acquisition and NAOqi APIs calls
- contains *link* nodes, *function* nodes for processing the messages' payloads, *change* nodes, *JSON* nodes, nodes for I/O streams and *pythonshell* nodes used to interact with python processes. *Exec* nodes also can be used for executing Python scripts
- developed in the context of neuromarketing
- for more intelligent connection between the BCI and the robot, a flow implementing fuzzy logic for featuring the personal EEG data will be designed. It will work in the background during the whole session with the user



❑ *EPOC-Pepper communications via Node-RED*

- exchanging data between nodes that belong to different flows in Node-RED
- global variables to send EEG data to Pepper
- *link-out* and *link-in* nodes to synchronize the timestamps of the sent and received data - the *link-out* node in the flow for BCI and the *link-in* node in the flow for the robot
- publishing and receiving messages was performed by *link-in* and *link-out* nodes in the broker flow2
- option for EPOC+ or Pepper to send value to a given topic from their flows directly via a Python script that calls *ALMemory APIs*
- advantage of the proposed solution is the lightening of the devices that have to be connected. There is no need a client software to publish-subscribe via a broker to be installed, such as *paho-MQTT for Eclipse mosquito*
- using of internal memory of the robot to share and collect data for topics allows publishing-subscribing to be done from another Node-RED instances or from external IDEs
- synchronization for collaborative tasks can be maintained

Conclusion

- We presented a novel approach for Brain-Robot communications in the IoT based on flow-based programing and cross-platform open-source server side runtime environment built in browser

- It uses the publish-subscribe communication model underlying Node-RED that allows the devices and services to exchange data by message passing in a network connecting “black boxes”

- The proposed approach can be applied for different BCI and robot things that have nodes in the Node-RED palette or are open-source programmable by Python or ROS

- In the future we plan to enhance the Pepper robot’s awareness of human emotion by integrating in the Pepper Python SDK the valence classification model by fuzzy inference on EEG data obtained by a BCI device

Thank You!

SVETOSLAV KREMENSKI