# Introduction

# Related Work

A software architecture for multimodal emotion recognition was described in Gonzales-Sanches et al. (2011). The core components of their architecture include sensing devices, specialist agents, a centre agent, and third-party systems. The sensing devices have a one-way communication to their specialist agent, so the specialist agent is receiving data from the sensing device, but the sensing device is not receiving any data from the specialist agent. Each specialist agent has a two-way communication with the centre agent, so the specialist agent can send data to and receive data from the centre agent. The purposes of the communication from the centre agent to a specialist agent include to coordinate sample rates across multiple devices. Similarly, the centre agent has a two-way communication with third-party systems.

Each specialist agent is uniquely designed around the sensing device. The specialist agent has multiple components which perform the following tasks: communicating with the sensor, preliminary processing or reformatting of the data, sending the data to the centre agent. Some manufacturers offer SDKs and/or APIs for one to communicate with the sensor via bluetooth, while others use USB ports. The specialist agent is responsible for reformatting the data so that it is sent in the desired format to the centre agent.

While there exists a specialist agent for each sensor, the architecture uses one centre agent. The centre agent is responsible for communicating with every specialist agent, processing the multimodal data, and communicating to the third-party software. The way in which the centre agent processes the multimodal data can vary by task, but always integrates the multiple data streams to update the current state. The centre agent communicates the current state to the third-party software at a frequency set within the architecture design.

Gonzales-Sanches et al. describe the challenges presented by vastly different sample rates for each sensor and present some options for overcoming these challenges. One example of overcoming the sampling differences is to record the multimodal data as a data table and continue to add a new row anytime any sensor gets new data; the sensor attributes that received new data will be updated in the new row and any sensor attributes that did not receive new data have their previous data included on the new row. The current state can be calculated at any time based on the newest row in the data table. This solution works best for continuity of data but can create misleading interpretations of the data if one sensor has not been updated for a relatively longer period than other sensors, because the interpretation assumes that the data indicated is accurate for that timestamp. Another solution is to adjust the sample rate from the sensors with the highest sample rate so that each sensor collects one sample every time the slowest sensor collects a sample. Both methods can be used successfully but have drawbacks, so it comes to the application of the data and the sensor-types when determining a solution.

(Insert TAG background here)

The University of Arizona’s TAG project’s ongoing research into robotic teaching agents is currently using the LEGO® MINDSTORMS® EV3 robot to drive across a classroom to assist students to interpret geometric shapes. The EV3 robot combines a Wi-Fi-capable EV3 Intelligent Brick to two medium motors each connected to a medium wheel on either side of the EV3 Intelligent Brick (**Figure X – EV3 Configuration**). The TAG project installed a python script onto the EV3 which allows one to send movement commands to the EV3 from a remote computer by communicating using MQTT protocols and leveraging the LEGO® MINDSTORMS® Education EV3 MicroPython SDK (Kumar 2021). The TAG project’s EV3 is designed to move upon receiving an MQTT message in the format of: “MoveTank {left speed} {right speed}”, where “{left speed}” and “{right speed}” are floating point values ranging from 0 to 100 and used to denote the percentage of max speed at which the left and right motors should turn the left and right wheels, respectively. For example, the command “MoveTank 50 50” would prompt the TAG EV3 to power both wheels at 50% max power, therefore resulting in a forward movement. Alternatively, the command “MoveTank 50, 0” would prompt the TAG EV3 to power the left wheel at 50% max power while the right wheel remains stationary, therefore resulting in a spinning motion centered around the right wheel.

# Methods and Procedures

The project team was interested in exploring the complex challenge of mapping sensing device data to movement commands for the EV3 using standalone sensors and multimodal sensors.

## EV3 Movement Commands

The team leveraged the software loaded on the TAG EV3 to receive MQTT messages in the form of “MoveTank {left speed} {right speed}” to move the robot, however, the existing movement commands were limited. The researcher enhanced the existing commands by developing the python script “sender\_functions\_TB” which has a variety of pre-formulated messages to send over MQTT for movement commands such as: go straight, stop, go straight for a preset amount of time, move with bi-speed controls, turn around 180-degrees, spin, and spin for a preset amount of time. This python script was developed to be imported by centre agents and to be used to communicate to the third-party software (EV3 robot) to facilitate additional movement commands.

## EEG Signals

The first sensing device the researchers chose was the EMOTIV Insight EEG brainwave detection sensor (“the Insight”). The Insight is a 5-channel EEG with a 1-to-2-minute preparation time and utilizes a proprietary 2.4GHz wireless BLE connection. The relatively fast preparation time and wireless connectivity made the Insight the ideal EEG for the researchers because it would work best for future deployment with children in a classroom setting if integrated with the TAG project. The Insight can stream the raw EEG data directly from the sensors, but it can also stream Performance Metrics (stress, engagement, interest, relaxation, focus, and excitement).

The Insight’s specialist agent architecture incorporates the Emotiv© Cortex API and an adaptation of the “cortex.py” and “sub\_data.py” python scripts from the Emotiv’s Cortex example public GitHub repository (Emotiv 2021). The modified python scripts allowed the researchers to subscribe to the Performance Metrics from the Insight, and the data was sent using MQTT protocol to the centre agent for further processing.

To first test the possibility of mapping the Insight’s data into movement commands for the EV3 robot, the researchers developed the python script, “int\_func\_EEG\_1d\_onMSG.py” to receive the Insight data sent over MQTT from the EEG specialist agent and send a command to the EV3 to move forward, stop, move backward, or stop continuously upon receipt of a message from the EEG specialist agent. The test script worked successfully, so the team built off that script to send movement commands to the EV3 based on the content of the Insight data – not just on the receipt of new data.

The first mapping of the Insight’s data into movement commands for the EV3 robot was utilized movements in one dimension so the robot moved forwards or backwards during any movements – both wheels were powered to always matching speeds. The speed and direction of the EV3 robot was dictated by the changes in the Insight’s relaxation Performance Metric from the previous sample to the newest. While the mapping successfully worked as intended, the researchers quickly noticed that the Insight’s Performance Metrics are difficult to control even if trying deliberately. The researchers decided that the Performance Metrics would likely work best when actively being collected, but only triggering movement commands when a metric exceeds a range instead of actively mapping the changes from sample to sample.

## Empatica E4 Signals

The second sensor the team integrated into the EV3 movement architecture was the Empatica E4 wristband. The E4 wristband is a medical-grade wearable device that offers real-time physiological data acquisition. Equipped with sensors to monitor blood volume pulse, galvanic skin response, 3-axis accelerometer, and skin temperature, the E4 sensor was an ideal choice for the variety of data streams accessible from the single device and therefore vast variety of applications and movement mapping options.

The E4 can connect to a computer using a BLE-Bluetooth dongle. Empatica offers an API for developers to stream the data for their own purposes. In order to do so, one must acquire an API key from Empatica, install the “Empatica Streaming Server”, and connect to the streaming server using Transmission Control Protocol/Internet Protocol (TCP/IP). The research team built off existing research from the University of Arizona’s SensorLab for streaming E4 data using in a python script but modified to send the data to a centre agent over MQTT (see python script “sub\_data\_E4.py” from the project GitHub).

The project team was interested in exploring the complex challenge of mapping sensing device data to movement commands for the EV3 using standalone sensors and multimodal sensors.

The researcher designed a specialist agent and centre agent for processing EEG signals and communicating movement commands to the EV3.

The project team selected the EMOTIV Insight EEG

# Results

# Conclusions

## References

Emotiv. (2021). “cortex-v2-example.” GitHub repository, https://github.com/Emotiv/cortex-v2-example/tree/master/python

Gonzalez-Sanchez, Javier & Chavez-Echeagaray, Maria-Elena & Atkinson, Robert & Burleson, Winslow. (2011). “ABE: An Agent-Based Software Architecture for A Multimodal Emotion Recognition Framework.” 10.1109/WICSA.2011.32.

Kumar, Rishav. (2021). “MQTTws-ev3.” GitLab repository, https://gitlab.com/rishavk1/mqttws-ev3