**Section 1 System Functionalities**

The system is designed to detect pre-learnt dance moves of an user, from the data from sensors attached on the user’s body. The data from sensors is sent from the Arduino Mega 2560 to the Raspberry Pi to analyze and predict the dance moves. The predicted dance moves are then sent to a remote evaluation server.

* 1. **Use cases**

**Use case:** Normal dance move

1. The user dances according to which move is displayed on screen.
2. The sensors from the user’s body sent the data to the system.
3. The system analyzes the data.
4. The system sends the predicted dance move to the evaluation server.
5. Server receives the predicted dance move and continues to the next one.
6. The screen displays the next dance move for the user.

Use case ends.

**Use case:** Final dance move

1. The user dances the last move of the evaluation period.
2. The sensors from the user’s body sent the data to the system.
3. The system analyzes the data.
4. The system predicted that the dance move is the final one.
5. The system terminates.

Use case ends.

* 1. **Feature lists**

- Arduino Mega 2560:

+ Receive data and filter noises from sensors.

+ Push data to the Raspberry Pi through UART, every 200ms.

- Raspberry Pi:

+ Receive data from the Arduino through UART.

+ Clean data before sending it to the Machine Learning (ML) module.

+ The ML module returns a predicted dance move, which will be sent to the remote evaluation server.

- Additional hardware:

+ A level shifter is used as a “bridge” to transfer data between the Arduino and RPi.

* 1. **Non-functional requirements**

1. In Raspbian on RPi, only the necessary processes are allowed to run, to reduce resource consumption.
2. The system should be able to be used ~1 hour continuously.
3. The system’s hardware should not limit the users’ body action.
4. The system’s hardware should not be loosened by users’ dancing.

**Section 4 Firmware & Communications Details**

**4.3. Protocol between Arduino Mega and Raspberry Pi**

The messages between two devices are sent through UART, with a baud rate of 9600 bps.

As the Arduino Mega 2560 runs at 16MHz, using baud rate generator formula, baud rate of 9600 bps is chosen since it is fast enough and only has the error rate of 0.2%, while the error rate of baud rate of 115200 bps is 3.7%.

On bootup, the Arduino will actively listen to a "start" message sent from the RPi, after the RPi finishes booting up. The Arduino then pushes data from sensors to the RPi, every 200ms, until it receives a "stop" message.

Raspberry Pi

Server

Communication Module

Arduino Mega 2560

Sensors

Machine Learning

**4.4. Communication between Raspberry Pi and the evaluation server**

Communication between the RPi and the server will be symmetrically encrypted with Advanced Encryption Standard (AES) in Cipher Block Chaining (CBC) mode. In CBC, the message is being XORed with the previous message or a random initialization vector if the message is the first one.

Using Socket, the server listens on one of its ports for messages from RPi. The messages are the predicted dance moves from the Machine Learning module.