**Remote Touchpad Protocol**

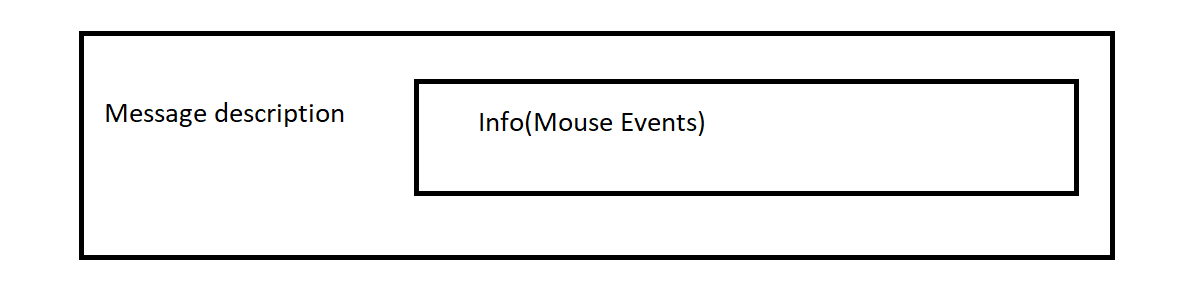
The connection is Bluetooth based. Before starting the first session, both devices, the PC (the server) and the smartphone (the client), should pair. The pairing should be initiated by the client.

To pair, the client requires the server’s MAC address. This will be obtained by displaying a QR code on the computer monitor which contains the Bluetooth adapter’s serial number(MAC). The client should prompt the user to scan the QR barcode.

Once paired, the client should connect to the server. The MAC address for connecting the devices should be acquired using the same method above, or by using a saved address from a previous session or from the pairing process.

To check the state of the connection, both sides, should send messages at intervals of 5 seconds after which they will await a response that acknowledges the message. The time out for the response is 5 seconds too. This way we will find out whether the connection is alive or not.

At the end of each session, the side which initiates the termination of the session should send a message that exclaims it.

The following describes the structure of the information sent on top of Bluetooth in our application:   
To make our project more modular, we have decided to split the information into two segments, one responsible for maintaining the connection and one for the purpose of our project, operating the mouse:

The first segment, we will name “Message Description”, and the second, “Info”.

The Message Description will determine whether the message represents Information about mouse events (in this case the info field will contain that information), a request for connectivity check, an acknowledgement of such request or an end of session message.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Message Type | Info length | Info |
| Description | Describes the type of the messag | The length of the info field | Contains the data of the second segment (In case it’s a mouse event message) |
| Size | One byte | One byte | At maximum, 255 bytes, at minimum, 0 bytes |
| Possible Values | 0 - Mouse events code | 0 <= x <=255 | Mouse Events [………] (See explanation of the second segment) |
| 1 – Connectivity check request code | 0 | [nothing] |
| 2 - Connectivity check acknowledgement code |
| 3 – Termination of session code |

Connectivity check request:  
Should be sent every 5 seconds. The sender should wait for a message code 2 response. The time out for the respond is 5 seconds.

Connectivity check acknowledgment:  
Should be sent when message code 1 is received.

Termination of session:  
Sent when one of the sides wants to end the session.

Mouse Event:  
Contains information about the mouse events that should be executed.  
Has the info field.  
Might contain not only one mouse event action but batches of them.

The second segment (The info field of Mouse Event Message):  
Information about the actions made by the touchpad is divided to small sets of data. Each set describes one action. Each set contains a value for the type of the action and data that contains parameters for each action (for example, the scroll action has a parameter that represents the magnitude of the scroll).

The Mouse Event message might contain

The structure the sets of data:  
[Action Code] | [Arguments]

* Action code - the action code that represents the type action described in this set.
* Arguments - the arguments that describe the details of the action.

The following chart describes the actions and their arguments:

|  |  |  |
| --- | --- | --- |
| Action Code | Action Type | Arguments |
| 0 | Mouse movement | dx, dy - Relative coordinates(these are signed values, positive for right/down and negative for left/up)  \*for negative values google up two’s component |
| 1 | Left button | Up/down - 0 for down, 1 for up, 2 for click (when the touchpad is just tapped we can’t describe it with button down/up, it’s a click ) |
| 2 | Right button | Up/down: 0 for down, 1 for up |
| 3 | Scroll | Data - represents the scale of the scroll (signed) |
| 4 | Zoom | Data - represents the sale of the scroll (signed) |

An example for a valid set of data that describe a mouse movement with the relative coordinates dx = 50 and dy = -35:

|  |  |  |  |
| --- | --- | --- | --- |
| Decimal (with sign) | 0 | 50 | -35(221) |
| Hex code | 0x00 | 0x32 | 0xDD |