COMMAND DESCRIPTION

Every command is sent via Bluetooth from PC/Mobile to the Master modules HC-05.

The master module receives the command and according to it takes specific action, it also broadcasts a relevant command set (if needed) to the slave modules for synchronization and movement.

Each **command is a string of characters**. Each **character can be an alphabet or a number**. Each character has some special meaning associated with it.

To know thoroughly about the commands go through the following sections of the code:

1) void listen_bluetooth() & void listen_RF()

These functions **give an insight about the first character** i.e. rf_rx_data[0]=bt_rx_data[0] which is received via wireless communication.

void listen_bluetooth() is **called** only if **module is master**. This receives data directly from user (PC/Mobile).

void listen_RF is **called** only if **module is slave**. Slave receives data from master module using RF(nRF24L01+) communication.

a => all modules ('a' means command for master as well as all slaves)

1 => command is for module number 1 (*1 is always master*)

n => command is for nth module

    n=2,3,4,5….

2) void loop()

This function **gives an insight about the second character** i.e. final_rx_data[1] which is received via wireless communication.

r   => snake   (**r**un sinusoidal)

e   => escape   (end all movements)

s   => set_angle   (sets angle manually)

h => hook_attach   (attach modules=hook)

n => not angle   (switches angle between two states -90⇔0⇔90)

w   => wheel   (run wheel)

Further characters of the command, i.e. **character number 3,4,5,6 and so on can be explained by looking at the individual functions** called **in** the loop function, when a specific second character is switched by the **switch case in the loop function**.

I) snake (if second character is r)

Snake motion can be observed in male direction (direction where the male face is facing) or in the female direction. (Forward or reverse)

The **3rd character** defines the direction in case of snake motion.

m => male direction

f => female direction

Eg: **arm**, **arf**, **1rm**, **1rf**, **2rm**, **2rf** ...

II) escape (if second character is e)

There is NO 3rd character for this command. The escape command for one specific or for all modules to stop is of 2 characters.

Eg: **ae**, **1e**, **2e**, **3e** ...

III) set_angle (if second character is s)

The **3rd character defines** the angle of which **hinge servo (male/female)** is to be set

m => male hinge servo

f => female hinge servo

The remaining characters(2 or 3) define the angle.

setting angle of **hinge servos** is accomplished by using the below commands

Eg: **1sm90**, **1sf90**, **1sm-90**, **1sf-90 ...**

IV) hook_attach (if second character is h)

The **3rd character** defines the servo of which **face** is to be altered

b => base

l => left

r => right

The **4ᵗʰ character** defines the **attaching/detaching**

0 => detach

1 => attach

2 => detach and push (maximum detach)

setting angle of **face servos (attaching/detaching)** is accomplished using the below commands

Eg: **ahb0**,**ahb1**, **1hb0**,**1hb1** …

V) NOT angle (if second character is n)

'NOT' angle. Basically switches angle between 2 states, either 0<->90 or 0<->-90.

The **3ʳᵈ character defines** the angle of which **hinge servo (male/female)** is to be not,

m => male hinge servo

f => female hinge servo

The remaining characters(2 or 3) define the angle.

Eg: **anm90**. **1nm-90**, **anf-90**, **2nf-90** …

VI) wheel (if second character is w)

The **3ʳᵈ character defines** either prepare or run

p=> prepare wheel

r=> run wheel

Eg: **awp**, **awr**

NOTE: **All strings/words written in this document which are underlined and bold are COMMANDS except the note statement.**