MMTS FUNCTION DOCUMENTATION

* uint8\_t data\_from\_iot(void) //checking data from UART ring buffer

Here checking for the data from Iot, that is data available or not.

Here we are extracting the name, mode, duration of time from json format.

NO arguments passing.

Return - 1: if data from Iot is true data then return 1 for successfully received true data.

Return - 0: if data from Iot is not true data then return 0 for successfully not received true data.

* void motor\_drive\_clockwise(void) // Clock wise rotation function

Driving motor for clock-wise direction (Default position).

Motor input 1 – SET

Motor input 2 – RESET

PWM is 40%.

NO arguments passing.

Not returning anything.

* void motor\_drive\_anticlockwise(void)//anti-clock wise rotation function

Driving motor for anti-clockwise direction (90 degree position).

Motor input 1 – RESET

Motor input 2 – SET

PWM is 25%.

NO arguments passing.

Not returning anything.

* Void timer\_start(void)//timer function

Timer will start running for given seconds from Iot.

Calling buzzer function to indicate the timer as started.

Time\_flag =1; set timer flag to indicate the timer is ON, and call laser detect function in super loop.

NO arguments passing.

Not returning anything.

* void Buzzer(void)//buzzer function

Buzzer used to indicate timer as stared.

And also used to indicate laser gun is shot.

NO arguments passing.

Not returning anything.

* void laser\_detect(void)//laser detecting function

Once timer as stared, then start detecting that laser gun is shot or not.

If laser gun is shot then move motor to default position after a delta time of 5 sec.

And stop the timer.

NO arguments passing.

Not returning anything.

* void HAL\_TIM\_PeriodElapsedCallback(TIM\_HandleTypeDef\*htim)//timer elapsed function

If timer is elapsed for given seconds and laser is not shot, then passing a message to Iot and RIS ECU that laser is not shot further to take the action.

NO arguments passing.

Not returning anything.

RIS FUNCTION DOCUMENTATION

* def on\_connect(client, userdata, flags, rc) :

Call back for when the client receives a connect response from the server.

If we lose the connection & reconnect then subscription will be renewed.

Passing 4 arguments: client, userdata, flag, RC

* def on\_message(client, userdata, msg):

Call back for when a publish msg is received from the server.

* def forward(x):

Function to move the motor in forward direction.

* def reverse(x):

Function to move the motor in reverse direction.