

```

InitMotorService:
{
    Init I/O ports;
    Init UART;
    Init state for state machine;
    For packet reception state machine, CurrentState is WAIT_FOR_7E;
    For packet processing state machine, CurrentState is UNPAIRED;
    For packet transmission state machine, CurrentState is WAIT_FOR_SEND;
}

```

```

RunMotorService:
{
    Processing state machine:
        UNPAIRED:
            if MI6 requires to pair with me:
                if current MI6 is different from last MI6, send pair
                acknowledge message
            }
            Else if acknowledge message is sent successfully
                Update state to be PAIRED
                Turn floating fan on
                Init COMM_TMR for 1 sec
            }
            Else if RE-PAIR TMR is time out
                Clear MI6 address

        PAIRED:
            if receiving control command:
                if un-pair is required, raise Unpair_Flag
                else, check popping command, thrust,
                orientation and brake, and response

                Send status message
                Init COMM_TMR for 1 sec

            Else if my balloon is popped
                Mind state is set to be controlled

            Else if status message is sent successfully
                if Mind state is controlled, update address of
                Last MI6
                Update state to be UNPAIRED
                Init Re-pair timer to be 10 secs

                Else if Unpair_Flag is raised
                Update state to be UNPAIRED
                Reset Unpair_Flag

            Else if COMM_TMR timeout
                Update state to be UNPAIRED

```

Reception state machine:

```
WAIT_FOR_7E:
    If 0x7E received
        Update state to be WAIT_FOR_MSB
        Init 100ms RECEIVING_TMR

WAIT_FOR_MSB:
    If MSB for package length received
        Update length data
        Update state to be WAIT_FOR_LSB
        Init 100ms RECEIVING_TMR
    Else if RECEIVING_TMR timeout
        Update state to be WAIT_FOR_7E

WAIT_FOR_LSB:
    If LSB for package length received
        Update length data
        Update state to be SUCK_UP_PACKET
        Init 100ms RECEIVING_TMR
    Else if RECEIVING_TMR timeout
        Update state to be WAIT_FOR_7E

SUCK_UP_PACKET:
    If Data received
        If not reaching package length
            Save it to data array
            Init 100ms RECEIVING_TMR
        Else,

            Check sum
            Translate received data based on
            protocol, and post event to processing
            state machine
    Else if RECEIVING_TMR timeout
        Update state to be WAIT_FOR_7E
```

Transmission state machine

```
WAIT_FOR_SEND:
    if required to send a package
        Construct package based on protocol
        Send the package to TX interrupt module
```