

InitPlayerBrainSM()

init pwm ports
configure pin 2 on port P to be an input

this function exists in PropellerModule.c, header for that module is included in PlayerBrainSM.c
Call Initialize_PropellerModule()

this function exists in PropellerModule.c, header for that module is included in PlayerBrainSM.c
Call Initialize_ServoModule()
reset propellers
set variable rightCall to 0

End InitPlayerBrainSM

this function is called in PlayerLiFKIM

Build_LiFKIM(char byte, int i)

fill i'th element of array called LiFKIMArray with byte

End Build_LiFKIM**Read_Team()**

read switch on pin 2 of port P
if switch is high
 green team selcted
Else
 red team is selected
End if

End Read_Team

this function is called in PlayerReceiveSM

Decode_Preamble

pull the address bytes from incoming packet and store in array called addressArray (*that lives in PlayerTransmitSM module*)

End Decode_Preamble**Decode_Message**

if first byte of RF Data (aka Header) is Reset
 Post RESET to PlayerBrainSM
 Post Event TIME_OUT to PlayerLiFKIM
 put propellers to inital position
End If

if Header indicates a Tag Out Message

 check for what color
 Call Build_TagOut () and send the color to PlayerLiFKIM module
 check for what location
 Call Build_TagOut and send the location to PlayerLiFKIM module
 Post Event TAG_CALL to PlayerLiFKIM - do this 4 times
 Post Event TAG_OUT to PlayerBrain

reset propellers
End If

if Header Indicates a Tag Detected Message
check for what color
Call Build_TagDetected() and send the color to PlayerLiFKIM module
check for what location
Call Build_TagDetected() and send the location to PlayerLiFKIM module
Post Event TAG_CALL to PlayerLiFKIM - do this 4 times
End If

If Header indicates a Pair Request
If team is green team
If request is for red team player
Post Event DENY_PAIR to PlayerBrainSM

Build_Response is a function used to pass array elements to PlayerTransmit.c
Call Build_Response() and send message Header byte to PlayerTransmit module
Call Build_Response() and send error byte to PlayerTransmit module
End if
End If

If team is red team
If request is for green team player
Post Event DENY_PAIR to PlayerBrainSM

Build_Response is a function used to pass array elements to PlayerTransmit.c
Call Build_Response() and send message Header byte to PlayerTransmit module
Call Build_Response() and send error byte to PlayerTransmit module
End if
Else If team number is wrong
do nothing
Else If Player is already Paired
Post DENY_PAIR Event to PlayerBrain

Build_Response is a function used to pass array elements to PlayerTransmit.c
Call Build_Response() and send message Header byte to PlayerTransmit module
Call Build_Response() and send error byte to PlayerTransmit module
End If

If team is green team
If request is for green team player
Post Event GOOD_CALL to PlayerBrainSM

Build_Response is a function used to pass array elements to PlayerTransmit.c
Call Build_Response() and send message Header byte to PlayerTransmit module
Call Build_Response() and send error byte to PlayerTransmit module

set variable rightCall to 1
Else if team is red team
If request is for red team player
Post Event GOOD_CALL to PlayerBrainSM

Build_Response is a function used to pass array elements to PlayerTransmit.c
Call Build_Response() and send message Header byte to PlayerTransmit module
Call Build_Response() and send error byte to PlayerTransmit module
set variable rightCall to 1
End If
End If

End If

If Header is request for control status from LiFKIM
If Player is in Wait_Link state and if rightCall variable is 1
Post Event ACCEPT_PAIR to PlayerBrain
Post Event START_ENGINE to PlayerLiFKIM
Else If
store up to 5 bytes of control data from LiFKIM

Build_Response is a function used to pass array elements to PlayerTransmit.c
Call Build_Response() and send message Header byte to PlayerTransmit module
Call Build_Response() and send LiFKIM status bytes to PlayerTransmit module

Call Steer Player()
End If

If Kick action was commanded
Call Kick() - which is in PropellerModule.
End If

Read Energy Level from LiFKIM
move Energy Level indicator in accord with info from LiFKIM

End If

Post Event GET_STATUS to PlayerLiFKIM
End If

End Decode_Message

RunPlayerBrainSM()

If CurrentState is Wait_Link
if ThisEvent is DENY_PAIR
Post Event COACH_CALL to PlayerTransmitSM
NextState to Wait_Link
end if

```
if ThisEvent is GOOD_CALL
  Post Event COACH_CALL to PlayerTransmitSM
  NextState is Wait_Link
end if
```

```
if ThisEvent is ACCEPT_PAIR
  Post Event CAN_TRANSMIT to PlayerTransmitSM
  NextState is Paired
  rightCall variable to 0
end if
```

```
if ThisEvent is RESET
  Post Event TIME_OUT to PlayerTransmitSM
  NextState to Wait_Link
end if
```

```
if ThisEvent is TAG_OUT
  Post Event TAG_OUT to PlayerTransmitSM
  NextState to Wait_Link
end if
```

End if

If CurrentState is Paired

```
If This Event is ES_TIMEOUT
  NextState to Wait_Link
  reset propellers
End If
```

```
If This Event is ALREADY_PAURED
  Post Event ALREADY_PAURED to PlayerTransmitSM
  NextState to Paired
End If
```

```
If This Event is RESET
  Post Event TAG_OUT to PlayerTransmitSM
  NextState to Wait_Link
End if
```

```
If This Event is TAG_OUT
  Post Event TAG_OUT to PlayerTransmitSM
  NextState to Wait_Link
End if
```

End If

CurrentState = NextState

End RunPlayerBrainSM