

ME218C Final Project, 2016
Team 6
Comm SM Pseudocode (PAC Side)

PAC Comm State Machine

Module Variables: myPriority, txData, rxData, EncKey, txPacketLen, rxPacketLen, currentTxByte, RecState, txAddrMSB, txAddrLSB, curEncByte, lastEncChecksum, curColor

InitCommSM

Takes nothing, returns nothing

Initialize UART
Set priority

End InitCommSM

PostCommSM

Takes parameter ES_Event ThisEvent, returns true/false Returns ES_PostToService with this service's priority and event

End PostCommSM

RunCommSM

Takes parameter ES_Event ThisEvent, returns ES_Event ReturnEvent

Comm state machine
 Waiting4Start state
 If you receive start delimiter, go to waiting for MSB state
 Waiting4MSB state
 if you receive empty byte, go to waiting for LSB state
 Waiting4LSB state
 If you get a byte
 go to receiving data state
 set packet len variable
 Receiving data state
 if you're at packet length
 check checksum
 if good, post event to appropriate SM
 else throw away packet
 go back to waiting2start state
 if not yet at length
 store byte in data array
 add new byte to checksum

End CommSM

sendPairPacket

Takes lobbyist Id, returns nothing

set packet len var
load pair packet values into txData array
Calculate Checksum
Send it!

End sendPairPacket

sendEncryptionPacket

Takes nothing, returns nothing

```
set packet len var
load encryption packet values into txData array
generate random values for key
Calculate Checksum
Send it!
```

End sendPairPacket

sendStatusPacket

Takes paired and decryption error bool, returns nothing

```
setPacketLen
set paired bit
load packet values into txData array
load last encrypted checksum
Calculate Checksum
Send it!
```

End sendStatusPacket

sendControlPacket

Takes forward int8, turn int8 and special int8, returns nothing

```
setPacketLen
set paired bit
load packet values into txData array
calculate control checksum
Calculate packet checksum
Send it!
```

End sendStatusPacket

getNextByte

Takes nothing, returns uint8

```
return next byte from txData array
```

End getNextByte

isTxComplete

Takes nothing, returns bool

```
check if we've transmitted the entire packet
```

End isTxComplete

calcChecksum

Takes nothing, returns uint8

```
add all the packet bytes
subtract from 0xFF
```

End calcChecksum

getRxDataAPIID

Takes nothing, returns uint8

return the last rxDataAPIID

End getRxDataAPIID

getPacketType

Takes nothing, returns uint8

if last packet was a receive
 return the protocol packet type

End getPacketType

getPairData

Takes nothing, returns uint8

if last packet was pair request
 return data byte

End getPairData

getStatusData

Takes nothing, returns uint8

if last packet was status packet
 return data byte

End getStatusData

setTxAddrToLastSender

Takes nothing, returns nothing

if last packet was receive
 store address

End setTxAddrToLastSender

calcCtrlChecksum

Takes nothing, returns nothing

Sum the control packet values and return

End calcCtrlChecksum

confirmMSGOrigin

Takes nothing, returns bool

check message origin against stored value

End confirmMSGOrigin

checkEncByte

Takes nothing, return bool

Check ctrl checksum, return true if the packet is a retransmission

End checkEncByte

resendLastPacket

Takes nothing, returns nothing

resent counter and resend the last packet

End resendLastPacket

setColor

Takes bool color, returns nothing

Set color module variable

End setColor