Pseudo Code for SendingCommSM

module variables: CurrentState, ReturnEvent, Byte2Send, StatusArray, ChangeCityArray, QueryArray

RunSendingCommSM

takes ES\_Event as input and returns ES\_Event

Set NextState to be CurrentState, assume no transitions

Initialize EntryEventKind to ES\_Entry

Set ReturnEvent to CurrentEvent, assume event will be propagated up

if CurrentState is WaitingInSend

if Event is ES\_SendCMD

Set Byte2Send to Event parameter

Change NextState to SendingAllBytes

Change MakeTransition flag to true

Set EntryEventKind to ES\_Entry for the transition

Set ReturnEvent to ES\_No\_Event to consume the event

endif

else if SendingAllBytes

Call DuringSendingAllBytes with CurrentEvent as input

Assign the return from the function to CurrentEvent

if Event is ES\_EOT

Initialize a 2ms timer to give time between two transactions

Set ReturnEvent to ES\_NO\_Event to consume the event

else if Event is ES\_Timeout and Event parameter is PacTimer

Set NextState to WaitingInSend

Change MakeTransition to be true

endif

endif

if MakeTransition is true

Set CurrentEvent to ES\_Exit

Call RunSendingComm with ES\_Exit

Set CurrentState now to NextState to transition

Call RunSendingComm with ES\_Enty

endif

return ReturnEvent

end RunSendingCommSM

StartSendingCommSM

takes ES\_Event as input and returns nothing

if event input is ES\_Entry

Set CurrentState = ENTRY\_STATE

endif

Call RunSendingCommSM with Event as input

return whatever the function call returns

end StartSendingCommSM

DuringSendingAllBytes

takes in ES\_Event and return ES\_Event

Set ReturnEvent to input event

if input event is ES\_Entry or ES\_Entry\_history

Write Byte2Send to TXFIFO register

Write 0 to TXFIFO register 4 times

else if input event is ES\_Exit

do nothing

else

if Event is ES\_EOT

Read from register RXFIFO and call StoreByte with this reading as input

Do same 5 times

endif

endif

return ReturnEvent

end DuringSendingAllBytes

EOT\_Response

takes nothing and returns nothing

Post ES\_EOT to CommSM to indicate end of transmission

end EOT\_Response

StoreByte

takes two uint8\_t: Reading containing the information to be stored, ByteNumber, which byte number the info corresponds to. Returns nothing.

Switch Byte2Send

StatusRequest case: Set Reading to StatusArray

ChangeCityRequest case: Set Reading to ChangeCityArray

QueryRequest case: Set Reading to QueryArray

end switch

end StoreByte

RR\_Read

takes nothing returns uint8\_t

return third element of QueryArray

end RR\_Read

RS\_Read

takes nothing returns uint8\_t

return fourth element of QueryArray

end RS\_Read

SS1\_Read

takes nothing returns uint8\_t

return third element of StatusArray

end SS1\_Read

SS2\_Read

takes nothing returns uint8\_t

return fourth element of StatusArray

end SS2\_Read

SS3\_Read

takes nothing returns uint8\_t

return fifth element of StatusArray

end SS3\_Read