

\*\*\*\*\*

### **Pseudo-code for Checkin SM (Lowest Level)**

States: GETTING\_FREQUENCY\_1, SENDING\_FREQUENCY\_1,

GETTING\_FREQUENCY\_2, SENDING\_FREQUENCY\_2

Events Posted: ES\_SEND\_FREQ, ES\_ACK, ES\_NACK, ES\_INACTIVE

### **RunCheckinM**

Takes ES\_Event CurrentEvent, returns ES\_NO\_EVENT

Set MakeTransition variable to false, because we are not making a transition currently

Set state type variable NextState to CurrentState

Set event type EntryEventKind to ES\_ENTRY (default to normal entry to new state)

Set event type ReturnEvent to ES\_NO\_EVENT, assuming no error

Switch (CurrentState)

Case GETTING\_FREQUENCY\_1

Execute During function for GETTING\_FREQUENCY\_1

If the event is active (not ES\_NO\_EVENT)

Switch (Event)

Case: ES\_SEND\_FREQ

Set NextState to SENDING\_FREQUENCY\_1

Set MakeTransition to true

Set ReturnEvent to ES\_NO\_EVENT

End Case

End Switch

End if

End Case

Case SENDING\_FREQUENCY\_1

Call the SENDING\_FREQUENCY\_1 during function

Set CurrentEvent to returned event from during function

If the CurrentEvent is active (not ES\_NO\_EVENT)

Switch (CurrentEvent)

Case ES\_ACK

Set next state to GETTING\_FREQUENCY\_2

Set MakeTransition to true

Set ReturnEvent to ES\_NO\_EVENT (consumed)

End Case

Case ES\_NACK

Set next state to GETTING\_FREQUENCY\_1

Set MakeTransition to true

```

        Set ReturnEvent to ES_NO_EVENT (consumed)
    End Case
    Case ES_INACTIVE
        Set next state to SENDING_FREQUENCY_1
        Set MakeTransition to false
        Set ReturnEvent to ES_NO_EVENT (consumed)
    End Case

    End Switch
End if
End Case
Case GETTING_FREQUENCY_2
    Execute During function for GETTING_FREQUENCY_2
    If the event is active (not ES_NO_EVENT)
        Switch (Event)
            Case: ES_SEND_FREQ
                Set NextState to SENDING_FREQUENCY_2
                Set MakeTransition to true
                Set ReturnEvent to ES_NO_EVENT
            End Case
        End Switch
    End if
End Case

Case SENDING_FREQUENCY_2
    Call the SENDING_FREQUENCY_2 during function
    Set CurrentEvent to returned event from during function
    If the CurrentEvent is active (not ES_NO_EVENT)
        Switch (CurrentEvent)
            Case ES_ACK
                Set next state to SENDING_FREQUENCY_2
                Set MakeTransition to false
                Set ReturnEvent to ES_NO_EVENT (consumed)
            End Case
            Case ES_NACK
                Set next state to GETTING_FREQUENCY_1
                Set MakeTransition to true
                Set ReturnEvent to ES_NO_EVENT (consumed)
            End Case
            Case ES_INACTIVE
                Set next state to SENDING_FREQUENCY_2
                Set MakeTransition to false
                Set ReturnEvent to ES_NO_EVENT (consumed)
            End Case
        End Switch
    End if
End Case

```

End Case

End Switch

End if

End Case

If MakeTransition is true (we are transitioning to a different state)

Set the CurrentEvent to ES\_EXIT

Call RunCheckinSM with CurrentEvent

Set CurrentState to NextState

Call RunCheckinSM with ES\_ENTRY event (start the entry function for the new state)

Endif

Return ReturnEvent

End RunCheckinSM

### **StartCheckinSM**

Takes ES\_EVENT Current Event, returns nothing

Initialize CurrentState to GETTING\_FREQUENCY\_1

Call RunCheckinSM with Current Event (ES\_ENTRY event)

### **DuringWaitingToStart**

Takes Event, returns Event

Do nothing

Return Event

### **DuringGettingFrequency1**

Takes event, returns event

If event is ES\_ENTRY or ES\_ENTRY\_HISTORY

Start the Hall Effect Input Capture interrupt

Else if event is ES\_EXIT

Reset the counters for Input Capture ISR/Hall Effect calculations

Else

Nothing

Endif

Return Event (this event is either an event that CheckinSM needs to handle, or ES\_NO\_EVENT if a lower level SM handled it)

### **DuringGettingFrequency2**

Takes event, returns event

If event is ES\_ENTRY or ES\_ENTRY\_HISTORY

    Start the Hall Effect Input Capture interrupt

Else if event is ES\_EXIT

    Reset the counters for Input Capture ISR/Hall Effect calculations

Else

    Nothing

Endif

Return Event (this event is either an event that CheckinSM needs to handle, or ES\_NO\_EVENT if a lower level SM handled it)

### **DuringSendingFrequency1**

Takes event, returns event

If event is ES\_ENTRY or ES\_ENTRY\_HISTORY

    Send a report (ES\_SEND\_REPORT) to the comm service

    Set the event parameter as the code of the period

    Post the event to Comm service

Else if event is ES\_EXIT

Else

    If an ES\_INACTIVE event is posted,

        Send a ES\_DRIVE\_CHECKIN event to MasterSM to figure out what the active area is

Endif

Return Event (this event is either an event that CheckinSM needs to handle, or ES\_NO\_EVENT if a lower level SM handled it)

### **DuringSendingFrequency2**

Takes event, returns event

If event is ES\_ENTRY or ES\_ENTRY\_HISTORY

    Send a report (ES\_SEND\_REPORT) to the comm service

    Set the event parameter as the code of the period

    Post the event to Comm service

Else if event is ES\_EXIT

Else

    If an ES\_ACK event is posted,

        Send a ES\_DRIVE\_SHOOT event to MasterSM to move to shoot

    Else if an ES\_INACTIVE event is posted,

        Send an ES\_DRIVE\_CHECKIN event to find out what the active area is

    Endif

Endif

Return Event (this event is either an event that CheckinSM needs to handle, or ES\_NO\_EVENT if a lower level SM handled it)