**Pseudo Code for Wire Checker Service**

**InitWireCheckerService**

*Takes 8-bit variable”Priority,” returns boolean true or false.*

Pass input parameter to static variable “MyPriority”

Start 1 ms wire checker timer

Return true

End

**PostWireCheckerService**

*Takes ES\_Event, returns boolean true or false.*

Return ES\_PostToService with static variable “MyPriority” and input parameter “ThisEvent”

End

**RunWireCheckerService**

*Takes ES\_Event, returns ES\_Event.*

Initialize ES\_Event type variable “ReturnEvent”

Set “ReturnEvent” equal to ES\_NO\_EVENT

If input event is a wire checker timeout

Call CheckWire function

Start 2 ms wire checker timer

Else if input event is ES\_Sweep4Wire

Set static variable “Sweeping4Wire” to 1

Set static variable “LastSeenRightWire” to 0

Set static variable “LastSeenLeftWire” to 0

Endif

Return “ReturnEvent”

End

**CheckWire**

*Takes nothing, returns nothing*.

Initialize float array “Results”

Initialize static 32-bit variable “wireLostCounter”

Initialize static 8-bit variable “LastWireState” to FOUND\_WIRE

If campaign has begun

Call ReadWireSensors function, store sensor readings in “Results” array

Store first value of “results” array static variable “CurrentAmplitudeLeft”

Store second value of “results” array static variable “CurrentAmpltidueRight”

If “CurrentAmplitudeLeft” is less than “LowerThreshold” and “CurrentAmplitudeRight” is less than “LowerThreshold”

Increment “wireLostCounter” by 1

Endif

If “LastWireState” is FOUND\_WIRE and “wireLostCounter” is greater than “WIRE\_LOST\_THRESHOLD”

Post ES\_WireLost to MasterSM

Set “LastWireState” to LOST\_WIRE

Else if “LastWireState” is LOST\_WIRE and “CurrentAmplitudeLeft” is greater than “UpperThreshold”

Reset “wireLostCounter” to zero

Post ES\_WireFound to MasterSM

Set “LastWireState” to FOUND\_WIRE

Else if “LastWireState” is FOUND\_WIRE and “CurrentAmplitudeLeft” is greater than “UpperThreshold”

Reset “WireLostCounter” to zero

Endif

If “Sweeping4Wire” is equal to 1

If “LastSeenRightWire” is zero and “CurrentAmplitudeRight” is greater than “UpperThreshold”

Set “LastSeenRightWire” to 1

Post ES\_FoundRightWire to MasterSM

Endif

If “LastSeenLeftWire” is zero and “CurrentAmplitudeLeft” is greater than “UpperThreshold”

Set “LastSeenLeftWire” to 1

Post ES\_FoundLeftWire to MasterSM

Endif

If “LastSeenLeftWire” is 1 and “LastSeenRightWire” is 1

Set “Sweeping4Wire” to zero

Endif

Endif

Endif

Return

End

**ReadWireSensors**

*Takes float array “returnArray,” returns nothing.*

Initialize static 32-bit array “temp”

Initialize static floats “LeftReading” and “RightReading” to zero

Call ADC\_MultiRead function with input of “temp”

Set “LeftReading” to 0.6\*(first value of “temp”) + 0.4\*LeftReading

Set “RightReading” to 0.6\*(scaled second value of “temp”) + 0.4\*RightReading

Set first value of “returnArray” to “LeftReading”

Set second value of “returnArray”to “RightReading”

End

**GetWireSensorVals**

*Takes float array “returnArray,” returns nothing.*

Set first value of “returnArray” to “CurrentAmplitudeLeft”

Set second value of “returnArray” to “CurrentAmplitudeRight”

End

**bothWiresFound**

*Takes nothing, returns boolean true or false.*

Return true if “LastSeenLeftWire” and “LastSeenRightWire” are both 1

Otherwise, return false

End