Pseudo Code for DetectCityFrequency

module variable: MyPriority, CurrentState, CityPeriod, ValidCityPeriod

ValidCityFreqs array of possible city frequencies

InitDetectCityFrequencyService

takes in uint8\_t Priority of this service, returns bool to indicate successful init

Set MyPriority to Priority passed int

Set CurrentState to InitCityFrequency

Post to DetectCityFrequency ES\_Init

return true

end InitDetectCityFrequencyService

PostDetectCityFrequency

takes in ES\_Event, returns bool

call ES\_PostToService with event as input and return whatever the function call returns

end PostDetectCityFrequency

RunDetectCityFrequencyService

takes ES\_Event and returns ES\_Event

Initialize ReturnEvent to ES\_NO\_EVENT assuming no errors

declare function variable whichCity

switch CurrentState

case InitCityFrequency:

Set CurrentState to CheckCityFrequency

Initialize timer for 2ms timeout

case CheckingCityFrequency

if Event is ES\_Timout and Event parameter is CityFrequencyTimer

call CheckCityPeriod

assign function call return to variable whichCity

if whichCity is not 16 (an invalid city)

Post ES\_AtCity to MasterSM and to CaptureCityService

endif

InitTimer again for 1ms

else if Event is Timeout and parameter is CityFrequencyOneShot

Post ES\_NotAtCity to MasterSM

Set ValidCityPeriod to 16

Set CandidateCityValue to 16

endif

end switch

end RunDetectCityFrequency

CheckCityPeriod

takes nothing and returns uint8\_t indicating which city period we are reading (16 for invalid city)

function variables: counter, CurrentCityPeriod, LastCityValue

Set CurrentCityPeriod to CityPeriod

for i from 0 to 15

if CityPeriod value is near ValidCityFreqs[i] within a tolerance

Set CandidateCityValue to i

endif

endfor

if CandidateCityValue is not the same as LastCityValue

set counter to 0

else

increase counter by 1

endif

if counter is greater than 4

set counter to 0

Set ValidCityPeriod to CandidateCityValue

return CandidateCityValue

endif

Set LastCityValue to CandidateCityValue

return 16 (if we reached here then city period reading is invalid)

end CheckCityPeriod

InputCaptureResponseCityFrequency

takes nothing and returns nothing

clear the interrupt flag

Stop CityFrequencyOneShot Timer

Set ThisCapture to Timer value of timer used in input capture for city period

Set CityPeriod to ThisCapture - LastCapture

Set LastCapture to ThisCapture

Start CityFrequencyOneShot Timer with 200ms timeout

end InputCaptureResponseCityFrequency

GetValidCityPeriod

takes nothing, returns uint8\_t

return ValidCityPeriod

end GetValidCityPeriod