Single Road:

Globals:

* Counter: int
* Increment(void): void
* Decrement(void): void
* ISR\_Movement(void):void

Counter: global variable counts number of entities (Vehicles) in the road at the moment.

Increment:

* Description: increment the global variable Counter by 1.
* #App.

Decrement:

* Description: Decrement the global variable Counter by 1.
* #App.

ISR\_Movement:

* Description: Depends on the interrupt source increment or decrement the counter
* Calls Decrement() or Increment().
* #HW

System Scope:

* Initialization(void): void.
* Maximum: int.
* Tolerance: int.
* GreatestIndex: int.
* getRoadCounters(void):void
* getGreatestRoadCounter(void):int
* checkTolerance(void):Bool.
* chackRedState(void):Bool.
* chackYellowState(void):Bool.
* InvertStates(void):void.
* TimeUp(int):void
* NormalSequence(void):void
* RedOn(int):void
* GeenOn(int): void
* YellowOn(int): void
* mainApp(void):void

Description:

Initialization(void): void

* Initialize Maximum[4].
* Initialize Tolerance[4].
* GreatestIndex

getRoadCounters(void):void

* Read all road counters from their global variables.

getGreatestRoadCounter(void):int

* Return the maximum road capacity index/ID.

checkTolerance(void):Bool.

* Compare the maximum value to the greatest counter + (greatest counter \* (tolerance/100).
* Return true if greatest is greater than maximum.

chackRedState(void):Bool.

* Retun true if the current state of the targeted road (GreatestIndex) is red.
* Else return false.

chackYellowState(void):Bool.

* Retun true if the current state of the targeted road (GreatestIndex) is yellow.
* Else return false.

InvertStates(void):void.

* Turn Greens to Yellow then Red.
* Turn Red to Green.

TimeUp(int):void

* Adding 30 second to the current counter.

NormalSequence(void):void

* Trun RedOn delay 20s
* Trun Yellow On delay 10s
* Trun Green On delay 30

RedOn(int):void

* Turn on the red light

GreenOn(int): void

* Turn on the green light.

YellowOn(int): void

* Turn on the yellow light.

mainApp(void):void

* Call Initialization()
* Call getRoadCounters()
* Update GreatestIndex by the return of the getGreatestRoadCounter()
* Call checkTolerance.
* Check if checkTolerance return:
  + True:
    - Check red State of the target road.
      * True: InvertStates()
        + Start timer from 0.
      * False:
        + If true : check yellow state:

TimeUp(30)

* + - * + If false:

Activate Green()

* + False:
    - Normal Sequence()
* Loop all previous forever.