'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' \*

' DECLARATIONS \*

' \*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CONSTANTS:

INITIALIZING=0

OPERATING=1

INITIAL\_BLOCK\_INDEX=1

MAX\_BLOCK=12 '\* Max number of blocks for LOOPs, etc

MIN\_BLOCK\_INDEX=1

MAX\_BLOCK\_INDEX=13

LOWEST\_LOOP\_BLOCK=1

HIGHEST\_LOOP\_BLOCK=8

HIGHEST\_LOOP\_PASSING\_BLOCK=11

INITIAL\_IR\_EOT\_DETECTOR=1

MAX\_IR\_EOT\_DETECTOR=2

MAX\_IR\_EOT\_DETECTOR\_INDEX=3

INITIAL\_TURNOUT\_INDEX=1

MAX\_TURNOUT=20

MAX\_TURNOUT\_INDEX=21

MANUAL\_CAB\_INDEX = 0

INITIAL\_CAB\_INDEX = 1

MAX\_CAB=3

MAX\_CAB\_INDEX=4

MAX\_CAB\_SPEED\_INDEX=12

MAX\_SPEED\_NOTCH\_INDEX=11

SPEED\_NOTCH\_INCREMENT=6

TURNOUT\_DIRECTION\_PRIMARY=off '\* Turnout Primary (green) position of turnout

TURNOUT\_DIRECTION\_SECONDARY=on '\* Turnout Secondary(red) position of turnout

TORTOISE=1 '\* Turnout Types

ATLAS=2

DETECTOR\_BLOCK\_OCCUPIED=true '\* For block occupancy

DETECTOR\_BLOCK\_VACANT=false '\* For block occupancy

BLOCK\_STATUS\_VACANT = 0

BLOCK\_STATUS\_OCCUPIED\_EAST = 1

BLOCK\_STATUS\_OCCUPIED\_WEST = 2

BLOCK\_STATUS\_VACATED\_EAST = 3

BLOCK\_STATUS\_VACATED\_WEST = 4

BLOCK\_STATUS\_VACATED\_DELTA = 2

BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA = 10

HOLDS\_NO\_HOLD\_DECLARED = -1

COLOR\_UNSELECTED\_SPEED=$RGB\_DFDFDF '\* Color for unselected speed buttons

COLOR\_MANUAL\_CAB=$RGB\_FF0000

COLOR\_CAB\_1=$RGB\_2492FF

COLOR\_CAB\_2=$RGB\_008000

COLOR\_CAB\_3=$RGB\_005EBB

COLOR\_BRAKE\_ON = Red

COLOR\_BRAKE\_OFF = Black

COLOR\_DIRECTION\_FORWARD = Green

COLOR\_DIRECTION\_REVERSE = Red

COLOR\_BLOCK\_VACANT = Black

COLOR\_BLOCK\_IDLE = $RGB\_C0C0C0

COLOR\_INCLUDE\_VACATED\_BLOCK = Yellow

COLOR\_DONT\_INCLUDE\_VACATED\_BLOCK = Green

ICON\_DIRECTION\_FORWARD = Arrow\_North

ICON\_DIRECTION\_REVERSE = Arrow\_South

ICON\_CAB\_DOES\_NOT\_HAVE\_BLOCK = Square

ICON\_CAB\_HAS\_BLOCK\_EASTBOUND = Arrow\_East

ICON\_CAB\_HAS\_BLOCK\_WESTBOUND = Arrow\_West

ICON\_CAB\_HAS\_BLOCK\_WITHOUT\_OCCUPANCY = Lock

SIGNAL\_BLOCK\_INDICATOR\_NOT\_OCCUPIED = "\*\*\*"

SIGNAL\_BLOCK\_INDICATOR\_OCCUPIED = "GGG"

SIGNAL\_BLOCK\_INDICATOR\_VACATED = "YYY"

SIGNAL\_BLOCK\_INDICATOR\_MANUAL\_HOLD = "xRR"

EXTRA\_BLOCK\_MAX\_INDEX = 4

EXTRA\_BLOCK\_STAGING\_LEAD = 1

EXTRA\_BLOCK\_STAGING\_STRAIGHT = 2

EXTRA\_BLOCK\_STAGING\_DIVERGING = 3

EXTRA\_BLOCK\_MINE = 4

BLOCK\_STAGING = 8

BLOCK\_MINE = 5

TURNOUT\_STAGING = 3

TURNOUT\_STAGING\_SPLIT = 18

TURNOUT\_MINE = 12

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' NETWORK MODULE DECLARATIONS

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'''''''''''

''''''''''' Power Board "1"

'''''''''''

SMARTCABS:

SmartCab\_01, 'Orange

SmartCab\_02, 'Green

SmartCab\_03, 'Brown

'''''''''''

''''''''''' Turnout Control Board "1"

'''''''''''

' YardMaster (Module “1”)

' CONTROLS:

'TurnoutBd1\_1YardMstr01\_t05, TurnoutBd1\_1YardMstr02\_t08, TurnoutBd1\_1YardMstr03\_t14

'TurnoutBd1\_1YardMstr04\_tbd, TurnoutBd1\_1YardMstr05\_t06, TurnoutBd1\_1YardMstr06\_t19

'TurnoutBd1\_1YardMstr07\_t20, TurnoutBd1\_1YardMstr08\_t12, TurnoutBd1\_1YardMstr09\_t07

'TurnoutBd1\_1YardMstr10\_tbd, TurnoutBd1\_1YardMstr11\_tbd, TurnoutBd1\_1YardMstr12\_spare

'TurnoutBd1\_1YardMstr13\_t13, TurnoutBd1\_1YardMstr14\_tbd, TurnoutBd1\_1YardMstr15\_tbd

'TurnoutBd1\_1YardMstr16\_spare

' Sentry (Module “2”)

' SENSORS: 'replaced SES\_TCa2\_ with TurnoutBd1\_2Sentry

'TurnoutBd1\_2Sentry01\_t05, TurnoutBd1\_2Sentry02\_t08, TurnoutBd1\_2Sentry03\_t14

'TurnoutBd1\_2Sentry04\_tbd, TurnoutBd1\_2Sentry05\_t06, TurnoutBd1\_2Sentry06\_t19

'TurnoutBd1\_2Sentry07\_t20, TurnoutBd1\_2Sentry08\_t12, TurnoutBd1\_2Sentry09\_t07

'TurnoutBd1\_2Sentry10\_tbd, TurnoutBd1\_2Sentry11\_tbd, TurnoutBd1\_2Sentry12\_spare

'TurnoutBd1\_2Sentry13\_t13, TurnoutBd1\_2Sentry14\_tbd, TurnoutBd1\_2Sentry15\_tbd

'TurnoutBd1\_2Sentry16\_spare

'''''''''''

''''''''''' Block Control board "1"

'''''''''''

' Train Brain Module "1"

CONTROLS:

BlockBrd1\_1TBrainCont01\_08, BlockBrd1\_1TBrainCont02\_01, BlockBrd1\_1TBrainCont03\_05, BlockBrd1\_1TBrainCont04\_09

SENSORS:

BlockBrd1\_1TBrainSens01\_8E#, BlockBrd1\_1TBrainSens02\_8W#, BlockBrd1\_1TBrainSens03\_1E#, BlockBrd1\_1TBrainSens04\_1W#

' Train Brain Module "2"

CONTROLS:

BlockBrd1\_2TBrainCont01\_08, BlockBrd1\_2TBrainCont02\_01, BlockBrd1\_2TBrainCont03\_05, BlockBrd1\_2TBrainCont04\_09

SENSORS:

BlockBrd1\_2TBrainSens01\_5E#, BlockBrd1\_2TBrainSens02\_5W#, BlockBrd1\_2TBrainSens03\_9E#, BlockBrd1\_2TBrainSens04\_9W#

'Dash-8 Module "3"

CONTROLS:

BlockBrd1\_3Dash01\_08, BlockBrd1\_3Dash02\_01, BlockBrd1\_3Dash03\_05, BlockBrd1\_3Dash04\_09,

BlockBrd1\_3Dash05\_spare, BlockBrd1\_3Dash06\_spare, BlockBrd1\_3Dash07\_spare, BlockBrd1\_3Dash08\_spare

'''''''''''

''''''''''' Atlas Control Board "1"

'''''''''''

'CONTROLS:

'Atlas1\_Switchman01\_TBDPrim, Atlas1\_Switchman02\_TBDSec, Atlas1\_Switchman03\_t19Prim,

'Atlas1\_Switchman04\_t19Sec, Atlas1\_Switchman05\_t20Prim, Atlas1\_Switchman06\_t20Sec,

'Atlas1\_Switchman07\_TBDPrim, Atlas1\_Switchman08\_TBDSec, Atlas1\_Switchman09\_TBDPrim,

‘Atlas1\_Switchman10\_TBDSec, Atlas1\_Switchman11\_TBDPrim, Atlas1\_Switchman12\_TBDSec,

'Atlas1\_Switchman13\_TBDPrim, Atlas1\_Switchman14\_TBDSec,

'Atlas1\_Switchman15\_Spare, Atlas1\_Switchman16\_Spare

'''''''''''

''''''''''' Block Control board "2"

'''''''''''

' Train Brain Module "1"

CONTROLS: BlockBrd2\_1TBrainCont01\_07, BlockBrd2\_1TBrainCont02\_11, BlockBrd2\_1TBrainCont03\_04, BlockBrd2\_1TBrainCont04\_12

SENSORS: BlockBrd2\_1TBrainSens01\_7E#, BlockBrd2\_1TBrainSens02\_7W#, BlockBrd2\_1TBrainSens03\_11E#, BlockBrd2\_1TBrainSens04\_11W#

' Train Brain Module "2"

CONTROLS: BlockBrd2\_2TBrainCont01\_07, BlockBrd2\_2TBrainCont02\_11, BlockBrd2\_2TBrainCont03\_04, BlockBrd2\_2TBrainCont04\_12

SENSORS: BlockBrd2\_2TBrainSens01\_4E#, BlockBrd2\_2TBrainSens02\_4W#, BlockBrd2\_2TBrainSens03\_12E#, BlockBrd2\_2TBrainSens04\_12W#

'Dash-8 Module "3"

CONTROLS: BlockBrd2\_3Dash01\_07, BlockBrd2\_3Dash02\_11, BlockBrd2\_3Dash03\_04, BlockBrd2\_3Dash04\_12,

BlockBrd2\_3Dash05\_spare, BlockBrd2\_3Dash06\_spare, BlockBrd2\_3Dash07\_spare, BlockBrd2\_3Dash08\_spare

'''''''''''

''''''''''' Atlas Control Board "2"

'''''''''''

'CONTROLS:

'Atlas2\_Switchman01\_TBDPrim, Atlas2\_Switchman02\_TBDSec, Atlas2\_Switchman03\_TBDPrim,

'Atlas2\_Switchman04\_TBDSec, Atlas2\_Switchman05\_TBDPrim, Atlas2\_Switchman06\_TBDSec,

'Atlas2\_Switchman07\_Spare, Atlas2\_Switchman08\_Spare, Atlas2\_Switchman09\_Spare, Atlas2\_Switchman10\_Spare,

'Atlas2\_Switchman11\_Spare, Atlas2\_Switchman12\_Spare, Atlas2\_Switchman13\_Spare, Atlas2\_Switchman14\_Spare,

'Atlas2\_Switchman15\_Spare, Atlas2\_Switchman16\_Spare

'''''''''''

''''''''''' Turnout Control Board "2"

'''''''''''

' YardMaster (Module “1”)

CONTROLS: 'replaced YMC\_TCb1\_ with TurnoutBd2\_1YardMaster

TurnoutBd2\_1YardMstr01\_x, TurnoutBd2\_1YardMstr02\_t02, TurnoutBd2\_1YardMstr03\_t01

TurnoutBd2\_1YardMstr04\_t04, TurnoutBd2\_1YardMstr05\_t09, TurnoutBd2\_1YardMstr06\_t10

TurnoutBd2\_1YardMstr07\_t16, TurnoutBd2\_1YardMstr08\_t17, TurnoutBd2\_1YardMstr09\_t03

TurnoutBd2\_1YardMstr10\_t18, TurnoutBd2\_1YardMstr11\_t15, TurnoutBd2\_1YardMstr12\_tbd

TurnoutBd2\_1YardMstr13\_t11, TurnoutBd2\_1YardMstr14\_tbd, TurnoutBd2\_1YardMstr15\_tbd

TurnoutBd2\_1YardMstr16\_spare

' Sentry (Module “2”)

SENSORS: 'replaced SES\_TCb2\_ with TurnoutBd2\_2Sentry

TurnoutBd2\_2Sentry01\_x, TurnoutBd2\_2Sentry02\_t02, TurnoutBd2\_2Sentry03\_t01

TurnoutBd2\_2Sentry04\_t04, TurnoutBd2\_2Sentry05\_t09, TurnoutBd2\_2Sentry06\_t10

TurnoutBd2\_2Sentry07\_t16, TurnoutBd2\_2Sentry08\_t17, TurnoutBd2\_2Sentry09\_t03

TurnoutBd2\_2Sentry10\_t18, TurnoutBd2\_2Sentry11\_t15, TurnoutBd2\_2Sentry12\_tbd

TurnoutBd2\_2Sentry13\_t11, TurnoutBd2\_2Sentry14\_tbd, TurnoutBd2\_2Sentry15\_tbd

TurnoutBd2\_2Sentry16\_spare

'''''''''''

''''''''''' Block Control board "3"

'''''''''''

' Train Brain Module "1"

CONTROLS: BlockBrd3\_1TBrainCont01\_02, BlockBrd3\_1TBrainCont02\_06, BlockBrd3\_1TBrainCont03\_10, BlockBrd3\_1TBrainCont04\_03

SENSORS: BlockBrd3\_1TBrainSens01\_2E#, BlockBrd3\_1TBrainSens02\_2W#, BlockBrd3\_1TBrainSens03\_6E#, BlockBrd3\_1TBrainSens04\_6W#

' Train Brain Module "2"

CONTROLS: BlockBrd3\_2TBrainCont01\_02, BlockBrd3\_2TBrainCont02\_06, BlockBrd3\_2TBrainCont03\_10, BlockBrd3\_2TBrainCont04\_03

SENSORS: BlockBrd3\_2TBrainSens01\_10E#, BlockBrd3\_2TBrainSens02\_10W#, BlockBrd3\_2TBrainSens03\_3E#, BlockBrd3\_2TBrainSens04\_3W#

'Dash-8 Module "3"

CONTROLS: BlockBrd3\_3Dash01\_02, BlockBrd3\_3Dash02\_06, BlockBrd3\_3Dash03\_10, BlockBrd3\_3Dash04\_03,

BlockBrd3\_3Dash05\_spare, BlockBrd3\_3Dash06\_spare, BlockBrd3\_3Dash07\_spare, BlockBrd3\_3Dash08\_spare

'''''''''''

''''''''''' IRD Detection Board "1"

'''''''''''

' Sentry "1"

SENSORS:

IRDDetectBd1\_Sentry01\_1W, IRDDetectBd1\_Sentry02\_1E, IRDDetectBd1\_Sentry03\_2W, IRDDetectBd1\_Sentry04\_2E,

IRDDetectBd1\_Sentry05\_3W, IRDDetectBd1\_Sentry06\_3E, IRDDetectBd1\_Sentry07\_4W, IRDDetectBd1\_Sentry08\_4E,

IRDDetectBd1\_Sentry09\_5W, IRDDetectBd1\_Sentry10\_5E, IRDDetectBd1\_Sentry11\_6W, IRDDetectBd1\_Sentry12\_6E,

IRDDetectBd1\_Sentry13\_7W, IRDDetectBd1\_Sentry14\_7E, IRDDetectBd1\_Sentry15\_8W, IRDDetectBd1\_Sentry16\_8E

'''''''''''

''''''''''' IRD Detection Board "2"

'''''''''''

' Sentry "1"

' SENSORS:

'IRDDetectBd2\_Sentry01, IRDDetectBd2\_Sentry02, IRDDetectBd2\_Sentry03, IRDDetectBd2\_Sentry04, 'IRDDetectBd2\_Sentry05, IRDDetectBd2\_Sentry06, IRDDetectBd2\_Sentry07, IRDDetectBd2\_Sentry08,

'IRDDetectBd2\_Sentry09, IRDDetectBd2\_Sentry10, IRDDetectBd2\_Sentry11, IRDDetectBd2\_Sentry12, 'IRDDetectBd2\_Sentry13, IRDDetectBd2\_Sentry14, IRDDetectBd2\_Sentry15, IRDDetectBd2\_Sentry16

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' VARIABLES

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

VARIABLES:

' Holds address of controls for assigning cabs to blocks

CAB\_AorB\_Pointer[MAX\_BLOCK\_INDEX]

CAB\_CorD\_Pointer[MAX\_BLOCK\_INDEX]

CAB\_ABorCD\_Pointer[MAX\_BLOCK\_INDEX]

' Holds addresses of Current Detector Sensors for each block

CD\_East\_Pointer[MAX\_BLOCK\_INDEX]

CD\_West\_Pointer[MAX\_BLOCK\_INDEX]

' Holds addresses of IRD Detector Sensors for each block

IRD\_East\_Pointer[MAX\_BLOCK\_INDEX]

IRD\_West\_Pointer[MAX\_BLOCK\_INDEX]

' Holds address for Turnout Machine controls

Turnout\_Pointer[MAX\_TURNOUT\_INDEX]

' Holds address of Sensors for Manual turnout buttons

Turnout\_Button\_Pointer[MAX\_TURNOUT\_INDEX]

' Holds address of Controls for Atlas turnouts

Atlas\_Turnout\_Pointer\_Primary[MAX\_TURNOUT\_INDEX]

Atlas\_Turnout\_Pointer\_Secondary[MAX\_TURNOUT\_INDEX]

Atlas\_Turnout\_LED\_Pointer\_Primary[MAX\_TURNOUT\_INDEX]

Atlas\_Turnout\_LED\_Pointer\_Secondary[MAX\_TURNOUT\_INDEX]

' Holds address of 3 smart cabs and their functions (cab[0] is placeholder for manual cab and functions)

Cab\_Pointer[MAX\_CAB\_INDEX]

Cab\_Speed\_Pointer[MAX\_CAB\_INDEX] '\* holds address for speed for each cab

Cab\_Direction\_Pointer[MAX\_CAB\_INDEX] '\* holds address for direction for each cab

Cab\_Momentum\_Pointer[MAX\_CAB\_INDEX] '\* Holds address formomentum for each cab

Cab\_Brake\_Pointer[MAX\_CAB\_INDEX] '\* Holds address for brake for each cab

' Indexes for loops

BlockIndex ' BIndex

SpeedIndex ' SIndex

OccupancyIndex

IR\_Detector\_Index 'LOOP counter for checking IR detectors status

CurrentSensorIndex 'LOOP counter for checking current sensors status

QueryIndex

CabAssignIndex

TurnoutIndex

CabIndex ' CIndex

NotchIndex

index

InitStatus

' Status holders

Turnout\_Status[MAX\_TURNOUT\_INDEX] 'holds status of each turnout

Block\_Status[MAX\_BLOCK\_INDEX] 'BStatus

Block\_Cab[MAX\_BLOCK\_INDEX] '\* holds cab assignment for each block

' OccupyEast[MAX\_BLOCK\_INDEX] '\* holds status of current sensors

' OccupyWest[MAX\_BLOCK\_INDEX] '\* holds status of current sensors

Turnout\_Type[MAX\_TURNOUT\_INDEX]

Dcc\_Cab\_Pointer[MAX\_CAB\_INDEX]

Previous\_Dcc\_Cab\_Speed[MAX\_CAB\_INDEX]

'\* Arrays to hold Grid coordinates for panel display

Block\_Grid[MAX\_BLOCK\_INDEX]

Turnout\_Grid[MAX\_TURNOUT\_INDEX]

IR\_Detector\_Grid\_E[MAX\_BLOCK\_INDEX]

IR\_Detector\_Grid\_W[MAX\_BLOCK\_INDEX]

IR\_EOT\_Detector\_Grid[MAX\_IR\_EOT\_DETECTOR\_INDEX]

Cab1\_Speed\_Grid[MAX\_CAB\_SPEED\_INDEX]

Cab1\_Brake\_Grid

Cab1\_F\_R\_Indication\_Grid

Cab1\_Direction\_Grid

Cab2\_Speed\_Grid[MAX\_CAB\_SPEED\_INDEX]

Cab2\_Brake\_Grid

Cab2\_F\_R\_Indication\_Grid

Cab2\_Direction\_Grid

Cab3\_Speed\_Grid[MAX\_CAB\_SPEED\_INDEX]

Cab3\_Brake\_Grid

Cab3\_F\_R\_Indication\_Grid

Cab3\_Direction\_Grid

cab0\_Block\_Grid[MAX\_BLOCK\_INDEX]

cab1\_Block\_Grid[MAX\_BLOCK\_INDEX]

cab2\_Block\_Grid[MAX\_BLOCK\_INDEX]

cab3\_Block\_Grid[MAX\_BLOCK\_INDEX]

cab0\_All\_Blocks\_Grid

cab1\_All\_Blocks\_Grid

cab2\_All\_Blocks\_Grid

cab3\_All\_Blocks\_Grid

'\* Arrays to hold Colors & Shapes for panel display

Cab\_Color[MAX\_CAB\_INDEX]

Block\_Color[MAX\_BLOCK\_INDEX]

Block\_Sprite[MAX\_BLOCK\_INDEX]

Train\_E\_Sprite[MAX\_BLOCK\_INDEX]

Train\_W\_Sprite[MAX\_BLOCK\_INDEX]

IR\_Detector\_Shape\_E[MAX\_BLOCK\_INDEX]

IR\_Detector\_Shape\_W[MAX\_BLOCK\_INDEX]

IR\_EOT\_Detector\_Shape[MAX\_IR\_EOT\_DETECTOR]

'\* Speed Control

Previous\_Cab\_Speed[MAX\_CAB\_INDEX]

Speed\_Notch[MAX\_SPEED\_NOTCH\_INDEX] '\* holds throttle speeds for each step

' Auto chasing

Block\_Status\_Grid[MAX\_BLOCK\_INDEX]

Held\_Block\_Grid[MAX\_BLOCK\_INDEX]

Block\_Signal\_Indicator[MAX\_BLOCK\_INDEX]

LAYOUT\_CONTROL\_INCLUDE\_VACATED\_BLOCK

Vacated\_Block\_Button

Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_MAX\_INDEX]

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' \*

' ACTIONS \*

' \*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ACTIONS:

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' SUB ROUTINES

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\* Indicate Block occupancy change on panel

' Occupied

SUB Block\_occupancy\_off(block\_Grid,Block\_Color)

$Color Block (block\_Grid) = black

$Draw Sprite (block\_Grid) = Lock in Block\_Color

ENDSUB

SUB Block\_occupancy\_on(block\_Grid,block\_color,sprite\_shape)

$Color Block(block\_Grid) = block\_color

$Draw Sprite(block\_Grid) = sprite\_shape in Block\_Color

ENDSUB

'\*\*\*\*\*\*\*\* Configures TrainBrain Controls to assign proper cab to designated block

' Also assigns Cab color to designated block for panel displays

SUB Configure\_Cab\_To\_Block(BNum, CNum)

IF CNum >= MAX\_CAB\_INDEX THEN Return

ELSEIF CNum = MANUAL\_CAB\_INDEX then

\*Cab\_AorB\_Pointer[BNum] = off

\*Cab\_ABorCD\_Pointer[BNum] = off

ELSEIF CNum = 1 then

\*Cab\_AorB\_Pointer[BNum] = on

\*Cab\_ABorCD\_Pointer[BNum] = off

ELSEIF CNum = 2 then

\*Cab\_CorD\_Pointer[BNum] = off

\*Cab\_ABorCD\_Pointer[BNum] = on

ELSEIF CNum = 3 then

\*Cab\_CorD\_Pointer[BNum] = on

\*Cab\_ABorCD\_Pointer[BNum] = on

ENDIF

Block\_Color[BNum] = Cab\_Color[CNum]

ENDSUB

'\*\*\*\*\*\*\*\* Assign selected cab to designated blocks)

' Calls subroutine to configure corresponding trainbrain controls

' Colors block sprite with selected cab color

' Colors selected block buttons with selected cab color and shape on panel

SUB Redraw\_Extra\_Blocks\_On\_Grid({Local} CabIndex)

' Staging

IF Block\_Status[BLOCK\_STAGING] = BLOCK\_STATUS\_OCCUPIED\_EAST AND Turnout\_Status[TURNOUT\_STAGING] = TURNOUT\_DIRECTION\_SECONDARY

OR Block\_Status[BLOCK\_STAGING] = BLOCK\_STATUS\_OCCUPIED\_WEST AND Turnout\_Status[TURNOUT\_STAGING] = TURNOUT\_DIRECTION\_SECONDARY

THEN

CabIndex = Block\_Cab[BLOCK\_STAGING]

$Color Block(Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_STAGING\_LEAD]) = Cab\_Color[CabIndex]

IF Turnout\_Status[TURNOUT\_STAGING\_SPLIT] = TURNOUT\_DIRECTION\_PRIMARY THEN

$Color Block(Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_STAGING\_STRAIGHT]) = Cab\_Color[CabIndex]

$Color Block(Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_STAGING\_DIVERGING]) = COLOR\_BLOCK\_IDLE

ELSE

$Color Block(Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_STAGING\_STRAIGHT]) = COLOR\_BLOCK\_IDLE

$Color Block(Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_STAGING\_DIVERGING]) = Cab\_Color[CabIndex]

ENDIF

ELSE

$Color Block(Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_STAGING\_LEAD]) = COLOR\_BLOCK\_IDLE

$Color Block(Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_STAGING\_STRAIGHT]) = COLOR\_BLOCK\_IDLE

$Color Block(Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_STAGING\_DIVERGING]) = COLOR\_BLOCK\_IDLE

ENDIF

' Mine

IF Block\_Status[BLOCK\_MINE] = BLOCK\_STATUS\_OCCUPIED\_EAST AND Turnout\_Status[TURNOUT\_MINE] = TURNOUT\_DIRECTION\_SECONDARY

OR Block\_Status[BLOCK\_MINE] = BLOCK\_STATUS\_OCCUPIED\_WEST AND Turnout\_Status[TURNOUT\_MINE] = TURNOUT\_DIRECTION\_SECONDARY

THEN

CabIndex = Block\_Cab[BLOCK\_STAGING]

$Color Block(Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_MINE]) = Cab\_Color[CabIndex]

ELSE

$Color Block(Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_MINE]) = COLOR\_BLOCK\_IDLE

ENDIF

$Color Block(Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_MINE]) = COLOR\_BLOCK\_IDLE

ENDSUB

SUB Redraw\_Cab\_Block\_On\_Plan({Local} BlockIndex, CabIndex, BlockStatus)

BlockIndex = 0

UNTIL BlockIndex >= MAX\_BLOCK\_INDEX QUICKLOOP

CabIndex = Block\_Cab[BlockIndex]

BlockStatus = Block\_Status[BlockIndex]

IF BlockStatus >= BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA THEN

BlockStatus = BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA -

ENDIF

' Show the block color

IF BlockStatus = BLOCK\_STATUS\_OCCUPIED\_EAST OR BlockStatus = BLOCK\_STATUS\_OCCUPIED\_WEST THEN

$Color Block(Block\_Grid[BlockIndex]) = Cab\_Color[CabIndex]

ELSE

$Color Block(Block\_Grid[BlockIndex]) = COLOR\_BLOCK\_VACANT

ENDIF

' Show the icon

IF BlockStatus = BLOCK\_STATUS\_OCCUPIED\_EAST THEN

$Draw Sprite(Block\_Grid[BlockIndex]) = Train\_E\_Sprite[BlockIndex] IN Cab\_Color[CabIndex]

ELSEIF BlockStatus = BLOCK\_STATUS\_OCCUPIED\_WEST THEN

$Draw Sprite(Block\_Grid[BlockIndex]) = Train\_W\_Sprite[BlockIndex] IN Cab\_Color[CabIndex]

ELSE

$Draw Sprite(Block\_Grid[BlockIndex]) = ICON\_CAB\_HAS\_BLOCK\_WITHOUT\_OCCUPANCY IN Cab\_Color[CabIndex]

ENDIF

BlockIndex = 1 +

ENDLOOP

ENDSUB

SUB Redraw\_Cab\_Block\_On\_Grid({Local} BlockIndex)

BlockIndex = 0

UNTIL BlockIndex >= MAX\_BLOCK\_INDEX QUICKLOOP

' Reset to gray

$Draw Sprite(Cab0\_Block\_Grid[BlockIndex]) = ICON\_CAB\_DOES\_NOT\_HAVE\_BLOCK IN COLOR\_UNSELECTED\_SPEED

$Draw Sprite(Cab1\_Block\_Grid[BlockIndex]) = ICON\_CAB\_DOES\_NOT\_HAVE\_BLOCK IN COLOR\_UNSELECTED\_SPEED

$Draw Sprite(Cab2\_Block\_Grid[BlockIndex]) = ICON\_CAB\_DOES\_NOT\_HAVE\_BLOCK IN COLOR\_UNSELECTED\_SPEED

$Draw Sprite(Cab3\_Block\_Grid[BlockIndex]) = ICON\_CAB\_DOES\_NOT\_HAVE\_BLOCK IN COLOR\_UNSELECTED\_SPEED

' Then find correct color and mark it

IF Block\_Cab[BlockIndex] = 0 THEN

$Draw Sprite(Cab0\_Block\_Grid[BlockIndex]) = Block\_Sprite[BlockIndex] IN Cab\_Color[0]

ELSEIF Block\_Cab[BlockIndex] = 1 THEN

$Draw Sprite(Cab1\_Block\_Grid[BlockIndex]) = Block\_Sprite[BlockIndex] IN Cab\_Color[1]

ELSEIF Block\_Cab[BlockIndex] = 2 THEN

$Draw Sprite(Cab2\_Block\_Grid[BlockIndex]) = Block\_Sprite[BlockIndex] IN Cab\_Color[2]

ELSEIF BLock\_Cab[BlockIndex] = 3 THEN

$Draw Sprite(Cab3\_Block\_Grid[BlockIndex]) = Block\_Sprite[BlockIndex] IN Cab\_Color[3]

ENDIF

BlockIndex = 1 +

ENDLOOP

ENDSUB

SUB Redraw\_Block\_Signal\_Indicator({Local} BlockIndex, BlockStatus)

BlockIndex = MIN\_BLOCK\_INDEX

UNTIL BlockIndex >= MAX\_BLOCK\_INDEX QUICKLOOP

BlockStatus = Block\_Status[BlockIndex]

IF BlockStatus >= BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA THEN

BlockStatus = BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA -

ENDIF

IF BlockStatus = BLOCK\_STATUS\_VACANT THEN

$Signal(Block\_Signal\_Indicator[BlockIndex]) = SIGNAL\_BLOCK\_INDICATOR\_NOT\_OCCUPIED

ELSEIF BlockStatus = BLOCK\_STATUS\_OCCUPIED\_EAST OR BlockStatus = BLOCK\_STATUS\_OCCUPIED\_WEST THEN

$Signal(Block\_Signal\_Indicator[BlockIndex]) = SIGNAL\_BLOCK\_INDICATOR\_OCCUPIED

ELSEIF BlockStatus = BLOCK\_STATUS\_VACATED\_EAST OR BlockStatus = BLOCK\_STATUS\_VACATED\_WEST THEN

$Signal(Block\_Signal\_Indicator[BlockIndex]) = SIGNAL\_BLOCK\_INDICATOR\_VACATED

ENDIF

IF Block\_Status[BlockIndex] >= BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA THEN

$Signal(Block\_Signal\_Indicator[BlockIndex]) = SIGNAL\_BLOCK\_INDICATOR\_MANUAL\_HOLD

ENDIF

BlockIndex = 1 +

ENDLOOP

ENDSUB

SUB Redraw\_Cab\_Block\_All(BlockIndex)

Redraw\_Cab\_Block\_On\_Grid()

Redraw\_Cab\_Block\_On\_Plan()

Redraw\_Block\_Signal\_Indicator()

Redraw\_Extra\_Blocks\_On\_Grid()

ENDSUB

SUB Assign\_Cab\_To\_Block(CIndex,BIndex)

Block\_Cab[BIndex]=CIndex

Configure\_Cab\_To\_Block(BIndex,CIndex)

Redraw\_Cab\_Block\_All(BIndex)

ENDSUB

SUB Assign\_Cab\_To\_All\_Blocks(CIndex, {Local} I)

I = INITIAL\_BLOCK\_INDEX

UNTIL I >= MAX\_BLOCK\_INDEX QUICKLOOP

Assign\_Cab\_To\_Block(CIndex, I)

I = 1 +

ENDLOOP

ENDSUB

SUB Calculate\_Next\_Westward\_Block(BlockIndex, BlockFromIndex)

\*BlockFromIndex = BlockIndex

\*BlockFromIndex = 1 -

IF \*BlockFromIndex < LOWEST\_LOOP\_BLOCK THEN \*BlockFromIndex = HIGHEST\_LOOP\_BLOCK + ENDIF

ENDSUB

SUB Calculate\_Next\_Eastward\_Block(BlockIndex, EastwardBlock, {Local} CalculatedBlock)

CalculatedBlock = BlockIndex

{{

if block = 12, return -1 (and set other places to handle -1 return

if block = 9, block = 1

if block = 10, block = 3

if block = 11, block = 7

block++

if block > 8, block -= 8

if block = 1 & turnout 5 = thrown, block = 9

if block = 3 & turnout 9 = thrown, block = 10

if block = 7 & turnout 1 = thrown, block = 11

return block

}}

IF CalculatedBlock = 12 THEN Return ENDIF

IF CalculatedBlock = 9 THEN CalculatedBlock = 1 ENDIF

IF CalculatedBlock = 10 THEN CalculatedBlock = 3 ENDIF

IF CalculatedBlock = 11 THEN CalculatedBlock = 7 ENDIF

CalculatedBlock = 1 +

IF CalculatedBlock> HIGHEST\_LOOP\_BLOCK THEN CalculatedBlock = HIGHEST\_LOOP\_BLOCK - ENDIF

IF CalculatedBlock = 1, \*Turnout\_Pointer[5] = TURNOUT\_DIRECTION\_SECONDARY THEN CalculatedBlock = 9 ENDIF

IF CalculatedBlock = 3, \*Turnout\_Pointer[9] = TURNOUT\_DIRECTION\_SECONDARY THEN CalculatedBlock = 10 ENDIF

IF CalculatedBlock = 7, \*Turnout\_Pointer[1] = TURNOUT\_DIRECTION\_SECONDARY THEN CalculatedBlock = 11 ENDIF

\*EastwardBlock = CalculatedBlock

' \*EastwardBlock= BlockIndex

' \*EastwardBlock= 1 +

' IF \*EastwardBlock > HIGHEST\_LOOP\_BLOCK THEN \*EastwardBlock = HIGHEST\_LOOP\_BLOCK - ENDIF

ENDSUB

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* LOOP through East and West current detectors to determine block occupancy

SUB Current\_Detector\_Triggered\_East(BlockIndex, {Local} BlockFromIndex, BlockToIndex, CurrentCab)

Block\_Sprite[BlockIndex] = ICON\_CAB\_HAS\_BLOCK\_EASTBOUND

IF Block\_Status[BlockIndex] >= BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA THEN

Block\_Status[BlockIndex] = BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA

ELSE

Block\_Status[BlockIndex] = BLOCK\_STATUS\_VACANT

ENDIF

Block\_Status[BlockIndex] = BLOCK\_STATUS\_OCCUPIED\_EAST +

Redraw\_Cab\_Block\_All(BlockIndex)

IF BlockIndex > HIGHEST\_LOOP\_PASSING\_BLOCK THEN RETURN ENDIF

' Determine where we are coming from

' Not used now, but previously pulled the cab forward (not needed with set-ahead cab)

BlockFromIndex = -1, Calculate\_Next\_Westward\_Block(BlockIndex, &BlockFromIndex)

' Determine where we are going to

' In future, base this on turnouts controlled and turnout setting

BlockToIndex = -1, Calculate\_Next\_Eastward\_Block(BlockIndex, &BlockToIndex)

IF Block\_Status[BlockToIndex] = BLOCK\_STATUS\_VACANT THEN

' If way ahead is not blocked, move forward and grab control of next block

CurrentCab = Block\_Cab[BlockIndex]

Assign\_Cab\_To\_Block(CurrentCab, BlockToIndex)

ELSE

' Halt train, put a hold on the next block so we get it when it's released

CurrentCab = Block\_Cab[BlockIndex]

\*Cab\_Pointer[CurrentCab].Brake = ON

Held\_Block\_Grid[BlockToIndex] = CurrentCab

ENDIF

ENDSUB

SUB Current\_Detector\_Triggered\_West(BlockIndex)

Block\_Sprite[BlockIndex] = ICON\_CAB\_HAS\_BLOCK\_WESTBOUND

Block\_Status[BlockIndex] = BLOCK\_STATUS\_OCCUPIED\_WEST

Assign\_Cab\_To\_Block(Block\_Cab[BlockIndex], BlockIndex, BLOCK\_STATUS\_OCCUPIED\_WEST)

ENDSUB

SUB Current\_Detector\_Stopped\_Triggering(BlockIndex, {Local} PreviousBlockIndex, NewBlockStatus, PreviousBlockStatus)

Block\_Sprite[BlockIndex] = Square

NewBlockStatus = Block\_Status[BlockIndex]

IF LAYOUT\_CONTROL\_INCLUDE\_VACATED\_BLOCK THEN

NewBlockStatus = BLOCK\_STATUS\_VACATED\_DELTA +

ELSE

NewBlockStatus = BLOCK\_STATUS\_VACANT

ENDIF

IF Block\_Status[BlockIndex] >= BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA THEN

NewBlockStatus = BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA +

ENDIF

Block\_Status[BlockIndex] = NewBlockStatus

Assign\_Cab\_To\_Block(Block\_Cab[BlockIndex], BlockIndex)

IF BlockIndex > HIGHEST\_LOOP\_PASSING\_BLOCK THEN RETURN ENDIF

PreviousBlockIndex = -1, Calculate\_Next\_Westward\_Block(BlockIndex, &PreviousBlockIndex)

PreviousBlockStatus = Block\_Status[PreviousBlockIndex]

IF Block\_Status[PreviousBlockIndex] >= BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA THEN

PreviousBlockStatus = BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA -

ENDIF

If PreviousBlockStatus = BLOCK\_STATUS\_VACATED\_EAST or PreviousBlockStatus = BLOCK\_STATUS\_VACATED\_WEST THEN

PreviousBlockStatus = BLOCK\_STATUS\_VACANT

ENDIF

IF Block\_Status[PreviousBlockIndex] >= BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA THEN

PreviousBlockStatus = BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA +

ENDIF

Block\_Status[PreviousBlockIndex] = PreviousBlockStatus

ENDSUB

SUB Release\_Hold\_On\_Block(BlockIndex, {Local} CurrentCab)

IF Held\_Block\_Grid[BlockIndex] = HOLDS\_NO\_HOLD\_DECLARED THEN RETURN ENDIF

CurrentCab = Held\_Block\_Grid[BlockIndex]

\*Cab\_Pointer[CurrentCab].Brake = Off

Held\_Block\_Grid[BlockIndex] = HOLDS\_NO\_HOLD\_DECLARED

Assign\_Cab\_To\_Block(CurrentCab, BlockIndex)

ENDSUB

'\*\*\*\*\*\*\*\* Assign selected speed to cab

SUB Assign\_Speed\_To\_Cab(CIndex,SIndex)

\*Cab\_Pointer[CIndex].Speed = Speed\_Notch[SIndex], 'Set Actual cab speed

ENDSUB

' \*\*\*\*\*\*\*Initialization Sub Routines \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* Initialize Cab Speed Button Notch Values

SUB Initialize\_Speed\_Index({Local} SIndex, CurrentSpeed)

CurrentSpeed = 0

SIndex = 0

UNTIL SIndex >= MAX\_SPEED\_NOTCH\_INDEX QUICKLOOP

Speed\_Notch[SIndex] = CurrentSpeed

CurrentSpeed = SPEED\_NOTCH\_INCREMENT +

SIndex = 1 +

ENDLOOP

ENDSUB

SUB Initialize\_Cab\_Status()

wait 1

\*Cab\_Brake\_Pointer[1]=on

Assign\_Speed\_To\_Cab(1,7)

\*Cab\_Direction\_Pointer[1]=forward

\*Cab\_Pointer[1].Momentum = 3

\*Cab\_Brake\_Pointer[2]=on

Assign\_Speed\_To\_Cab(2,7)

\*Cab\_Direction\_Pointer[2]=forward

\*Cab\_Pointer[2].Momentum = 3

\*Cab\_Brake\_Pointer[3]=on

Assign\_Speed\_To\_Cab(3,7)

\*Cab\_Direction\_Pointer[3]=forward

\*Cab\_Pointer[3].Momentum = 3

ENDSUB

SUB Initialize\_Cab\_Speed\_And\_Direction\_On\_Display({Local} I)

I = 0

UNTIL I >= MAX\_CAB\_SPEED\_INDEX QUICKLOOP

$color Track(cab1\_Speed\_Grid[I]) = COLOR\_UNSELECTED\_SPEED

$color Track(cab2\_Speed\_Grid[I]) = COLOR\_UNSELECTED\_SPEED

$color Track(cab3\_Speed\_Grid[I]) = COLOR\_UNSELECTED\_SPEED

I = 1 +

ENDLOOP

ENDSUB

SUB Initialize\_Cab\_Direction\_Display()

$Color Track(cab1\_Brake\_Grid)=red

$Draw Sprite(cab1\_F\_R\_Indication\_grid)=Arrow\_North in Green

$Color Track(cab2\_Brake\_Grid)=red

$Draw Sprite(cab2\_F\_R\_Indication\_grid)=Arrow\_North in Green

$Color Track(cab3\_Brake\_Grid)=red

$Draw Sprite(cab3\_F\_R\_Indication\_grid)=Arrow\_North in Green

ENDSUB

SUB Initialize\_Detect\_Initial\_Blocks({Local} CIndex, BIndex)

CIndex = INITIAL\_CAB\_INDEX

BIndex = INITIAL\_BLOCK\_INDEX

UNTIL BIndex >= MAX\_BLOCK\_INDEX QUICKLOOP

IF \*CD\_East\_Pointer[BIndex]=on then

Block\_Cab[BIndex]=CIndex

Configure\_Cab\_To\_Block(BIndex,CIndex)

Assign\_Cab\_To\_Block(CIndex, BIndex)

CIndex = 1+

ELSE

'Block\_Sprite[BIndex]=square

Assign\_Cab\_To\_Block(CIndex, BIndex)

'Block\_Status[BIndex]=BLOCK\_STATUS\_VACANT

ENDIF

BIndex=+

ENDLOOP

' Reloop, to throw actual logic triggering a block

BIndex = INITIAL\_BLOCK\_INDEX

UNTIL BIndex >= MAX\_BLOCK\_INDEX QUICKLOOP

IF \*CD\_East\_Pointer[BIndex] = On THEN Current\_Detector\_Triggered\_East(BIndex) ENDIF

IF \*CD\_West\_Pointer[BIndex] = On THEN Current\_Detector\_Triggered\_West(BIndex) ENDIF

BIndex=+

ENDLOOP

ENDSUB

SUB Initialize\_Set\_All\_Turnouts\_To\_Primary\_Direction({Local} TIndex)

TIndex = INITIAL\_TURNOUT\_INDEX

UNTIL TIndex >= MAX\_TURNOUT\_INDEX LOOP

$Switch(TurnOut\_Grid[TIndex])=TURNOUT\_DIRECTION\_PRIMARY 'Throw Panel Display

\*Turnout\_Pointer[TIndex]=TURNOUT\_DIRECTION\_PRIMARY ' Throw power to Tortoise/LED or Atlas LED only

IF Turnout\_Type[TIndex]=ATLAS THEN

\*Atlas\_Turnout\_Pointer\_Primary[TIndex]=PULSE 0.25 'throw power to Atlas Turnout

ENDIF

Turnout\_Status[TIndex]=TURNOUT\_DIRECTION\_PRIMARY

TIndex = 1+

ENDLOOP

ENDSUB

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' INITIALIZATION

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WHEN InitStatus=INITIALIZING do '(All lines must end in a comma to continue the initialization chain)

wait 2

LAYOUT\_CONTROL\_INCLUDE\_VACATED\_BLOCK = False

' set address pointers to cab selection controllers

'' board "1",

CAB\_AorB\_Pointer[8]=&BlockBrd1\_1TBrainCont01\_08,CAB\_AorB\_Pointer[1]=&BlockBrd1\_1TBrainCont02\_01,

CAB\_AorB\_Pointer[5]=&BlockBrd1\_1TBrainCont03\_05, CAB\_AorB\_Pointer[9]=&BlockBrd1\_1TBrainCont04\_09,

CAB\_CorD\_Pointer[8]=&BlockBrd1\_2TBrainCont01\_08,CAB\_CorD\_Pointer[1]=&BlockBrd1\_2TBrainCont02\_01,

CAB\_CorD\_Pointer[5]=&BlockBrd1\_2TBrainCont03\_05, CAB\_CorD\_Pointer[9]=&BlockBrd1\_2TBrainCont04\_09,

CAB\_ABorCD\_Pointer[8]=&BlockBrd1\_3Dash01\_08, CAB\_ABorCD\_Pointer[1]=&BlockBrd1\_3Dash02\_01,

CAB\_ABorCD\_Pointer[5]=&BlockBrd1\_3Dash03\_05, CAB\_ABorCD\_Pointer[9]=&BlockBrd1\_3Dash04\_09,

' SPARE=&BlockBrd1\_3Dash05, SPARE=&BlockBrd1\_3Dash06, SPARE=&BlockBrd1\_3Dash07, SPARE=&BlockBrd1\_3Dash08,

'' board "2",

CAB\_AorB\_Pointer[7]=&BlockBrd2\_1TBrainCont01\_07,CAB\_AorB\_Pointer[11]=&BlockBrd2\_1TBrainCont02\_11,

CAB\_AorB\_Pointer[4]=&BlockBrd2\_1TBrainCont03\_04, CAB\_AorB\_Pointer[12]=&BlockBrd2\_1TBrainCont04\_12,

CAB\_CorD\_Pointer[7]=&BlockBrd2\_2TBrainCont01\_07,CAB\_CorD\_Pointer[11]=&BlockBrd2\_2TBrainCont02\_11,

CAB\_CorD\_Pointer[4]=&BlockBrd2\_2TBrainCont03\_04, CAB\_CorD\_Pointer[12]=&BlockBrd2\_2TBrainCont04\_12,

CAB\_ABorCD\_Pointer[7]=&BlockBrd2\_3Dash01\_07, CAB\_ABorCD\_Pointer[11]=&BlockBrd2\_3Dash02\_11,

CAB\_ABorCD\_Pointer[4]=&BlockBrd2\_3Dash03\_04, CAB\_ABorCD\_Pointer[12]=&BlockBrd2\_3Dash04\_12,

' SPARE=&BlockBrd2\_3Dash05, SPARE=&BlockBrd2\_3Dash06, SPARE=&BlockBrd2\_3Dash07, SPARE=&BlockBrd2\_3Dash08,

'' board "3",

CAB\_AorB\_Pointer[2]=&BlockBrd3\_1TBrainCont01\_02,CAB\_AorB\_Pointer[6]=&BlockBrd3\_1TBrainCont02\_06,

CAB\_AorB\_Pointer[10]=&BlockBrd3\_1TBrainCont03\_10, CAB\_AorB\_Pointer[3]=&BlockBrd3\_1TBrainCont04\_03,

CAB\_CorD\_Pointer[2]=&BlockBrd3\_2TBrainCont01\_02,CAB\_CorD\_Pointer[6]=&BlockBrd3\_2TBrainCont02\_06,

CAB\_CorD\_Pointer[10]=&BlockBrd3\_2TBrainCont03\_10, CAB\_CorD\_Pointer[3]=&BlockBrd3\_2TBrainCont04\_03,

CAB\_ABorCD\_Pointer[2]=&BlockBrd3\_3Dash01\_02, CAB\_ABorCD\_Pointer[6]=&BlockBrd3\_3Dash02\_06,

CAB\_ABorCD\_Pointer[10]=&BlockBrd3\_3Dash03\_10, CAB\_ABorCD\_Pointer[3]=&BlockBrd3\_3Dash04\_03,

' SPARE=&BlockBrd3\_3Dash05, SPARE=&BlockBrd3\_3Dash06, SPARE=&BlockBrd3\_3Dash07, SPARE=&BlockBrd3\_3Dash08,

' set address pointers to Current Detector Sensors

'' board "1"

CD\_East\_Pointer[8]=&BlockBrd1\_1TBrainSens01\_8E, CD\_West\_Pointer[8]=&BlockBrd1\_1TBrainSens02\_8W,

CD\_East\_Pointer[1]=&BlockBrd1\_1TBrainSens03\_1E, CD\_West\_Pointer[1]=&BlockBrd1\_1TBrainSens04\_1W,

CD\_East\_Pointer[5]=&BlockBrd1\_2TBrainSens01\_5E, CD\_West\_Pointer[5]=&BlockBrd1\_2TBrainSens02\_5W,

CD\_East\_Pointer[9]=&BlockBrd1\_2TBrainSens03\_9E, CD\_West\_Pointer[9]=&BlockBrd1\_2TBrainSens04\_9W,

'' board "2"

CD\_East\_Pointer[7]=&BlockBrd2\_1TBrainSens01\_7E, CD\_West\_Pointer[7]=&BlockBrd2\_1TBrainSens02\_7W,

CD\_East\_Pointer[11]=&BlockBrd2\_1TBrainSens03\_11E, CD\_West\_Pointer[11]=&BlockBrd2\_1TBrainSens04\_11W,

CD\_East\_Pointer[4]=&BlockBrd2\_2TBrainSens01\_4E, CD\_West\_Pointer[4]=&BlockBrd2\_2TBrainSens02\_4W,

CD\_East\_Pointer[12]=&BlockBrd2\_2TBrainSens03\_12E, CD\_West\_Pointer[12]=&BlockBrd2\_2TBrainSens04\_12W,

'' board "3"

CD\_East\_Pointer[2]=&BlockBrd3\_1TBrainSens01\_2E, CD\_West\_Pointer[2]=&BlockBrd3\_1TBrainSens02\_2W,

CD\_East\_Pointer[6]=&BlockBrd3\_1TBrainSens03\_6E, CD\_West\_Pointer[6]=&BlockBrd3\_1TBrainSens04\_6W,

CD\_East\_Pointer[10]=&BlockBrd3\_2TBrainSens01\_10E, CD\_West\_Pointer[10]=&BlockBrd3\_2TBrainSens02\_10W,

CD\_East\_Pointer[3]=&BlockBrd3\_2TBrainSens03\_3E, CD\_West\_Pointer[3]=&BlockBrd3\_2TBrainSens04\_3W,

' set address pointers to IRD Detector Sensors

'' board "1"

IRD\_East\_Pointer[1]=&IRDDetectBd1\_Sentry02\_1E, IRD\_West\_Pointer[1]=&IRDDetectBd1\_Sentry01\_1W,

IRD\_East\_Pointer[2]=&IRDDetectBd1\_Sentry04\_2E, IRD\_West\_Pointer[2]=&IRDDetectBd1\_Sentry03\_2W,

IRD\_East\_Pointer[3]=&IRDDetectBd1\_Sentry06\_3E, IRD\_West\_Pointer[3]=&IRDDetectBd1\_Sentry05\_3W,

IRD\_East\_Pointer[4]=&IRDDetectBd1\_Sentry08\_4E, IRD\_West\_Pointer[4]=&IRDDetectBd1\_Sentry07\_4W,

IRD\_East\_Pointer[5]=&IRDDetectBd1\_Sentry10\_5E, IRD\_West\_Pointer[5]=&IRDDetectBd1\_Sentry09\_5W,

IRD\_East\_Pointer[6]=&IRDDetectBd1\_Sentry12\_6E, IRD\_West\_Pointer[6]=&IRDDetectBd1\_Sentry11\_6W,

IRD\_East\_Pointer[7]=&IRDDetectBd1\_Sentry14\_7E, IRD\_West\_Pointer[7]=&IRDDetectBd1\_Sentry13\_7W,

IRD\_East\_Pointer[8]=&IRDDetectBd1\_Sentry16\_8E, IRD\_West\_Pointer[8]=&IRDDetectBd1\_Sentry15\_8W

' set address pointers to Turnout\_Controls

'Turnout\_Pointer[5]=&TurnoutBd1\_1YardMstr01\_t05,

'Turnout\_Pointer[8]=&TurnoutBd1\_1YardMstr02\_t08,

'Turnout\_Pointer[14]=&TurnoutBd1\_1YardMstr03\_t14,

'TBD Turnout\_Pointer[atlas\_TBD]=&TurnoutBd1\_1YardMstr04\_TBD,

'Turnout\_Pointer[6]=&TurnoutBd1\_1YardMstr05\_t06,

'Turnout\_Pointer[19]=&TurnoutBd1\_1YardMstr06\_t19,

'Turnout\_Pointer[20]=&TurnoutBd1\_1YardMstr07\_t20,

'Turnout\_Pointer[12]=&TurnoutBd1\_1YardMstr08\_t12,

'Turnout\_Pointer[7]=&TurnoutBd1\_1YardMstr09\_t07,

'TBD Turnout\_Pointer[atlas\_TBD]=&TurnoutBd1\_1YardMstr10\_TBD,

'TBD Turnout\_Pointer[atlas\_TBD]=&TurnoutBd1\_1YardMstr11\_TBD,

'SPARE Turnout\_Pointer[Spare]=&TurnoutBd1\_1YardMstr12\_spare,

'Turnout\_Pointer[13]=&TurnoutBd1\_1YardMstr13\_t13,

'TBD Turnout\_Pointer[atlas TBD]=&TurnoutBd1\_1YardMstr14\_TBD,

'TBD Turnout\_Pointer[atlas TBD]=&TurnoutBd1\_1YardMstr15\_TBD,

'SPARE Turnout\_Pointer[Spare]=&TurnoutBd1\_1YardMstr16\_spare,

'DEFECTIVE? Turnout\_Pointer[X]=&TurnoutBd2\_1YardMstr01\_x,

Turnout\_Pointer[2]=&TurnoutBd2\_1YardMstr02\_t02,

Turnout\_Pointer[1]=&TurnoutBd2\_1YardMstr03\_t01,

Turnout\_Pointer[4]=&TurnoutBd2\_1YardMstr04\_t04,

Turnout\_Pointer[9]=&TurnoutBd2\_1YardMstr05\_t09,

Turnout\_Pointer[10]=&TurnoutBd2\_1YardMstr06\_t10,

'NIS Turnout\_Pointer[16]=&TurnoutBd2\_1YardMstr07\_t16,

'NIS Turnout\_Pointer[17]=&TurnoutBd2\_1YardMstr08\_t17,

Turnout\_Pointer[3]=&TurnoutBd2\_1YardMstr09\_t03,

Turnout\_Pointer[18]=&TurnoutBd2\_1YardMstr10\_t18,

'NIS Turnout\_Pointer[15]=&TurnoutBd2\_1YardMstr11\_T15,

'TBD Turnout\_Pointer[atlas TBD]=&TurnoutBd2\_1YardMstr12\_TBD

'NIS Turnout\_Pointer[11]=&TurnoutBd2\_1YardMstr13\_t11

'TBD Turnout\_Pointer[atlas TBD]=&TurnoutBd2\_1YardMstr14\_TBD

'TBD Turnout\_Pointer[atlas TBD]=&TurnoutBd2\_1YardMstr15\_TBD

'SPARE TurnoutBd2\_1YardMstr16,

' set address pointers to Turnout Push Button Sensors

'Turnout\_Button\_Pointer[5]=&TurnoutBd1\_2Sentry01\_t05,

'Turnout\_Button\_Pointer[8]=&TurnoutBd1\_2Sentry02\_t08,

'Turnout\_Button\_Pointer[14]=&TurnoutBd1\_2Sentry03\_t14,

'TBD Turnout\_Button\_Pointer[atlas\_TBD]=&TurnoutBd1\_2Sentry04\_TBD,

'Turnout\_Button\_Pointer[6]=&TurnoutBd1\_2Sentry05\_t06,

'Turnout\_Button\_Pointer[19]=&TurnoutBd1\_2Sentry06\_t19,

'Turnout\_Button\_Pointer[20]=&TurnoutBd1\_2Sentry07\_t20,

'Turnout\_Button\_Pointer[12]=&TurnoutBd1\_2Sentry08\_t12,

'Turnout\_Button\_Pointer[7]=&TurnoutBd1\_2Sentry09\_t07,

'TBD Turnout\_Button\_Pointer[atlas\_TBD]=&TurnoutBd1\_2Sentry10\_TBD,

'TBD Turnout\_Button\_Pointer[atlas\_TBD]=&TurnoutBd1\_2Sentry11\_TBD,

'SPARE Turnout\_Button\_Pointer[Spare]=&TurnoutBd1\_2Sentry12\_spare,

'Turnout\_Button\_Pointer[13]=&TurnoutBd1\_2Sentry13\_t13,

'TBD Turnout\_Button\_Pointer[atlas\_TBD]=&TurnoutBd1\_2Sentry14\_TBD,

'TBD Turnout\_Button\_Pointer[atlas\_TBD]=&TurnoutBd1\_2Sentry15\_TBD,

' SPARE Turnout\_Button\_Pointer[spare]=&TurnoutBd1\_2Sentry16\_Spare,

'DEFECTIVE? Turnout\_Button\_Pointer[X]=&TurnoutBd2\_2Sentry01\_x,

Turnout\_Button\_Pointer[2]=&TurnoutBd2\_2Sentry02\_t02,

Turnout\_Button\_Pointer[1]=&TurnoutBd2\_2Sentry03\_t01,

Turnout\_Button\_Pointer[4]=&TurnoutBd2\_2Sentry04\_t04,

Turnout\_Button\_Pointer[9]=&TurnoutBd2\_2Sentry05\_t09,

Turnout\_Button\_Pointer[10]=&TurnoutBd2\_2Sentry06\_t10,

'NIS Turnout\_Button\_Pointer[16]=&TurnoutBd2\_2Sentry07\_t16,

'NIS Turnout\_Button\_Pointer[17]=&TurnoutBd2\_2Sentry08\_t17,

Turnout\_Button\_Pointer[3]=&TurnoutBd2\_2Sentry09\_t03,

Turnout\_Button\_Pointer[18]=&TurnoutBd2\_2Sentry10\_t18,

' NIS Turnout\_Button\_Pointer[15]=&TurnoutBd2\_2Sentry11\_t15,

'TBD Turnout\_Button\_Pointer[Atlas TBD]=&TurnoutBd2\_2Sentry12\_TBD,

'NIS Turnout\_Button\_Pointer[11]=&TurnoutBd2\_2Sentry13\_t11,

'TBD Turnout\_Button\_Pointer[Atlas TBD]=&TurnoutBd2\_2Sentry14\_TBD,

'TBD Turnout\_Button\_Pointer[Atlas TBD]=&TurnoutBd2\_2Sentry15\_TBD,

'SPARE Turnout\_Button\_Pointer[Spare]=&TurnoutBd2\_2Sentry16\_Spare,

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas1\_Switchman01\_TBDPrim,

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas1\_Switchman02\_TBDSec,

'Atlas\_Turnout\_Pointer\_Primary[19]=&Atlas1\_Switchman03\_t19Prim,

'Atlas\_Turnout\_Pointer\_Secondary[19]=&Atlas1\_Switchman04\_t19Sec,

'Atlas\_Turnout\_Pointer\_Primary[20]=&Atlas1\_Switchman05\_t20Prim,

'Atlas\_Turnout\_Pointer\_Secondary[20]=&Atlas1\_Switchman06\_t20Sec,

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas1\_Switchman07\_TBDPrim,

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas1\_Switchman08\_TBDSec,

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas1\_Switchman09\_TBDPrim,

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas1\_Switchman10\_TBDSec,

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas1\_Switchman11\_TBDPrim,

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas1\_Switchman12\_TBDSec,

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas1\_Switchman13\_TBDPrim,

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas1\_Switchman14\_TBDSec,

'SPARES Atlas1\_Switchman15\_Spare, Atlas1\_Switchman16\_Spare

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas2\_Switchman01\_TBDPrim,

'TBD Atlas\_Turnout\_Pointer\_Secondary[TBD]=&Atlas2\_Switchman02\_TBDSec,

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas2\_Switchman03\_TBDPrim,

'TBD Atlas\_Turnout\_Pointer\_Secondary[TBD]=&Atlas\_Turnout\_Pointer\_Secondary[TBD]=&Atlas2\_Switchman04\_TBDSec,

'TBD Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas2\_Switchman05\_TBDPrim,

'TBD Atlas\_Turnout\_Pointer\_Secondary[TBD]=&Atlas\_Turnout\_Pointer\_Primary[TBD]=&Atlas2\_Switchman06\_TBDSec,

'SPARES Atlas2\_Switchman07\_Spare, Atlas2\_Switchman08\_Spare,

'SPARES Atlas2\_Switchman09\_Spare, Atlas2\_Switchman10\_Spare,

'SPARES Atlas2\_Switchman11\_Spare, Atlas2\_Switchman12\_Spare,

'SPARES Atlas2\_Switchman13\_Spare, Atlas2\_Switchman14\_Spare,

'SPARES Atlas2\_Switchman15\_Spare, Atlas2\_Switchman16\_Spare

' set address pointers to Cab Controls and Functions

Cab\_Pointer[1]=&SmartCab\_01,

Cab\_Pointer[2]=&SmartCab\_02,

Cab\_Pointer[3]=&SmartCab\_03,

Cab\_Speed\_Pointer[1]=&SmartCab\_01.speed,

Cab\_Speed\_Pointer[2]=&SmartCab\_02.speed,

Cab\_Speed\_Pointer[3]=&SmartCab\_03.speed,

Cab\_Direction\_Pointer[1]=&SmartCab\_01.Direction,

Cab\_Direction\_Pointer[2]=&SmartCab\_02.Direction,

Cab\_Direction\_Pointer[3]=&SmartCab\_03.Direction,

Cab\_Brake\_Pointer[1]=&SmartCab\_01.Brake,

Cab\_Brake\_Pointer[2]=&SmartCab\_02.Brake,

Cab\_Brake\_Pointer[3]=&SmartCab\_03.Brake,

Cab\_Momentum\_Pointer[1]=&SmartCab\_01.Momentum,

Cab\_Momentum\_Pointer[2]=&SmartCab\_02.Momentum,

Cab\_Momentum\_Pointer[3]=&SmartCab\_03.Momentum,

' set DCC cab pointers

Dcc\_Cab\_Pointer[1] = &CAB\_1,

Dcc\_Cab\_Pointer[2] = &CAB\_2,

Dcc\_Cab\_Pointer[3] = &CAB\_3,

'\*\* Initialize panel locations of IR Block Detectors - one at each end of block

IR\_Detector\_Grid\_E[1]=(46,14,4), IR\_Detector\_Grid\_E[2]=(27,26,4), IR\_Detector\_Grid\_E[3]=(7,18,4), IR\_Detector\_Grid\_E[4]=(24,2,4)

IR\_Detector\_Grid\_E[5]=(50,17,4), IR\_Detector\_Grid\_E[6]=(24,29,4), IR\_Detector\_Grid\_E[7]=(4,15,4), IR\_Detector\_Grid\_E[8]=(37,6,4)

IR\_Detector\_Grid\_E[9]=(45,14,4), IR\_Detector\_Grid\_E[10]=(8,19,4), IR\_Detector\_Grid\_E[11]=(6,15,4), 'IR\_Detector\_Grid\_E[12]=

IR\_Detector\_Grid\_W[1]=(33,26,4), IR\_Detector\_Grid\_W[2]=(7,22,4), IR\_Detector\_Grid\_W[3]=(21,2,4), IR\_Detector\_Grid\_W[4]=(50,13,4)

IR\_Detector\_Grid\_W[5]=(27,29,4), IR\_Detector\_Grid\_W[6]=(4,21,4), IR\_Detector\_Grid\_W[7]=(30,6,4), IR\_Detector\_Grid\_W[8]=(46,10,4)

IR\_Detector\_Grid\_W[9]=(34,24,4), IR\_Detector\_Grid\_W[10]=(21,3,4), IR\_Detector\_Grid\_W[11]=(32,7,4), 'IR\_Detector\_Grid\_W[12]=

IR\_EOT\_Detector\_Grid[0]=(16,5,4), IR\_EOT\_Detector\_Grid[1]=(16,4,4)

'\*\* Initialize panel locations of tortoise controls

Turnout\_Grid[1]=(4,18,4), Turnout\_Grid[2]=(9,12,4), Turnout\_Grid[3]=(34,6,4), Turnout\_Grid[4]=(35,6,4),

Turnout\_Grid[5]=(46,11,4), Turnout\_Grid[6]=(46,16,4), Turnout\_Grid[7]=(45,20,4), Turnout\_Grid[8]=(29,26,4),

Turnout\_Grid[9]=(7,19,4),Turnout\_Grid[10]=(20,2,4), Turnout\_Grid[11]=(23,2,4), Turnout\_Grid[12]=(50,16,4),

Turnout\_Grid[13]=(50,20,4), Turnout\_Grid[14]=(33,24,4), Turnout\_Grid[15]=(7,16,4), Turnout\_Grid[16]=(8,15,4),

Turnout\_Grid[17]=(4,23,4),Turnout\_Grid[18]=(31,4,4),Turnout\_Grid[19]=(51,15,4),Turnout\_Grid[20]=(52,14,4),

'\*\* Initialize panel location of track blocks and sprite locations of corresponding sprites

Block\_Grid[1]=(42,26,4), Block\_Grid[2]=(15,26,4), Block\_Grid[3]=(7,4,4), Block\_Grid[4]=(38,2,4)

Block\_Grid[5]=(42,29,4), Block\_Grid[6]=(15,29,4), Block\_Grid[7]=(13,6,4), Block\_Grid[8]=(42,6,4)

Block\_Grid[9]=(42,24,4), Block\_Grid[10]=(8,5,4), Block\_Grid[11]=(13,7,4), Block\_Grid[12]=(24,12,4),

'\*\* Initialize sprite shapes to overlay on assign cab pushbuttons

Train\_E\_Sprite[1]=Arrow\_west, Train\_W\_Sprite[1]=Arrow\_east, Train\_E\_Sprite[2]=Arrow\_west, Train\_W\_Sprite[2]=Arrow\_east

Train\_E\_Sprite[3]=Arrow\_north, Train\_W\_Sprite[3]=Arrow\_south, Train\_E\_Sprite[4]=Arrow\_east, Train\_W\_Sprite[4]=Arrow\_west

Train\_E\_Sprite[5]=Arrow\_west, Train\_W\_Sprite[5]=Arrow\_east, Train\_E\_Sprite[6]=Arrow\_west, Train\_W\_Sprite[6]=Arrow\_east

Train\_E\_Sprite[7]=Arrow\_East, Train\_W\_Sprite[7]=Arrow\_west, Train\_E\_Sprite[8]=Arrow\_east, Train\_W\_Sprite[8]=Arrow\_west

Train\_E\_Sprite[9]=Arrow\_west, Train\_W\_Sprite[9]=Arrow\_east, Train\_E\_Sprite[10]=Arrow\_north, Train\_W\_Sprite[10]=Arrow\_south

Train\_E\_Sprite[11]=Arrow\_east, Train\_W\_Sprite[11]=Arrow\_west, Train\_E\_Sprite[12]=Arrow\_east, Train\_W\_Sprite[12]=Arrow\_west

'\*\* Initialize sprite shapes of IR Block Detectors - one at each end of block

IR\_Detector\_Shape\_E[1]=Train\_South, IR\_Detector\_Shape\_E[2]=Train\_West, IR\_Detector\_Shape\_E[3]=Train\_North, IR\_Detector\_Shape\_E[4]=Train\_East

IR\_Detector\_Shape\_E[5]=Train\_South, IR\_Detector\_Shape\_E[6]=Train\_West, IR\_Detector\_Shape\_E[7]=Train\_North, IR\_Detector\_Shape\_E[8]=Train\_East

IR\_Detector\_Shape\_E[9]=Train\_South, IR\_Detector\_Shape\_E[10]=Train\_North, IR\_Detector\_Shape\_E[11]=Train\_North

IR\_Detector\_Shape\_W[1]=Train\_West, IR\_Detector\_Shape\_W[2]=Train\_North, IR\_Detector\_Shape\_W[3]=Train\_East, IR\_Detector\_Shape\_W[4]=Train\_South

IR\_Detector\_Shape\_W[5]=Train\_West, IR\_Detector\_Shape\_W[6]=Train\_North, IR\_Detector\_Shape\_W[7]=Train\_East, IR\_Detector\_Shape\_W[8]=Train\_South

IR\_Detector\_Shape\_W[9]=Train\_West, IR\_Detector\_Shape\_W[10]=Train\_East, IR\_Detector\_Shape\_W[11]=Train\_East

'IR\_Dector\_Shape\_E[12]=

'IR\_Dector\_Shape\_W[12]=

IR\_EOT\_Detector\_Shape[0]=Train\_East, IR\_EOT\_Detector\_Shape[1]=Train\_East

'\*\* Defines turnout Types

Turnout\_Type[1]=TORTOISE, Turnout\_Type[2]=TORTOISE, Turnout\_Type[3]=TORTOISE, Turnout\_Type[4]=TORTOISE,

Turnout\_Type[5]=TORTOISE, Turnout\_Type[6]=TORTOISE, Turnout\_Type[7]=TORTOISE, Turnout\_Type[8]=TORTOISE,

Turnout\_Type[9]=TORTOISE, Turnout\_Type[10]=TORTOISE, Turnout\_Type[11]=TORTOISE, Turnout\_Type[12]=TORTOISE,

Turnout\_Type[13]=TORTOISE, Turnout\_Type[14]=TORTOISE, Turnout\_Type[15]=TORTOISE, Turnout\_Type[16]=TORTOISE,

Turnout\_Type[17]=TORTOISE, Turnout\_Type[18]=TORTOISE,Turnout\_Type[19]=ATLAS, Turnout\_Type[20]=ATLAS,

'\*\* Initialize colors to indicate cab assignments on blocks and sprites

Cab\_Color[0]=COLOR\_MANUAL\_CAB, Cab\_Color[1]=COLOR\_CAB\_1, Cab\_Color[2]=COLOR\_CAB\_2, Cab\_Color[3]=COLOR\_CAB\_3,

'\*\* Initialize Cab Speed Pushbutton locations

Cab1\_Speed\_Grid[10]=(59,4,4), Cab1\_Speed\_Grid[9]=(59,5,4), Cab1\_Speed\_Grid[8]=(59,6,4), Cab1\_Speed\_Grid[7]=(59,7,4),

Cab1\_Speed\_Grid[6]=(59,8,4), Cab1\_Speed\_Grid[5]=(59,9,4), Cab1\_Speed\_Grid[4]=(59,10,4), Cab1\_Speed\_Grid[3]=(59,11,4),

Cab1\_Speed\_Grid[2]=(59,12,4), Cab1\_Speed\_Grid[1]=(59,13,4), Cab1\_Speed\_Grid[0]=(59,14,4),

Cab1\_Brake\_Grid=(59,16,4),

Cab1\_F\_R\_Indication\_Grid=(59,18,4),

Cab1\_Direction\_Grid=(59,18,4),

Cab2\_Speed\_Grid[10]=(61,4,4), Cab2\_Speed\_Grid[9]=(61,5,4), Cab2\_Speed\_Grid[8]=(61,6,4), Cab2\_Speed\_Grid[7]=(61,7,4),

Cab2\_Speed\_Grid[6]=(61,8,4), Cab2\_Speed\_Grid[5]=(61,9,4), Cab2\_Speed\_Grid[4]=(61,10,4), Cab2\_Speed\_Grid[3]=(61,11,4),

Cab2\_Speed\_Grid[2]=(61,12,4), Cab2\_Speed\_Grid[1]=(61,13,4), Cab2\_Speed\_Grid[0]=(61,14,4),

Cab2\_Brake\_Grid=(61,16,4),

Cab2\_F\_R\_Indication\_Grid=(61,18,4),

Cab2\_Direction\_Grid=(61,18,4),

Cab3\_Speed\_Grid[10]=(63,4,4), Cab3\_Speed\_Grid[9]=(63,5,4), Cab3\_Speed\_Grid[8]=(63,6,4), Cab3\_Speed\_Grid[7]=(63,7,4),

Cab3\_Speed\_Grid[6]=(63,8,4), Cab3\_Speed\_Grid[5]=(63,9,4), Cab3\_Speed\_Grid[4]=(63,10,4), Cab3\_Speed\_Grid[3]=(63,11,4),

Cab3\_Speed\_Grid[2]=(63,12,4), Cab3\_Speed\_Grid[1]=(63,13,4), Cab3\_Speed\_Grid[0]=(63,14,4),

Cab3\_Brake\_Grid=(63,16,4),

Cab3\_F\_R\_Indication\_Grid=(63,18,4),

Cab3\_Direction\_Grid=(63,18,4),

'\*\* Initialize Cab to Block Pushbutton locations

cab0\_Block\_Grid[1]=(57,21,4), cab0\_Block\_Grid[2]=(57,22,4), cab0\_Block\_Grid[3]=(57,23,4), cab0\_Block\_Grid[4]=(57,24,4),

cab0\_Block\_Grid[5]=(57,25,4), cab0\_Block\_Grid[6]=(57,26,4), cab0\_Block\_Grid[7]=(57,27,4), cab0\_Block\_Grid[8]=(57,28,4),

cab0\_Block\_Grid[9]=(57,29,4), cab0\_Block\_Grid[10]=(57,30,4), cab0\_Block\_Grid[11]=(57,31,4), cab0\_Block\_Grid[12]=(57,32,4),

cab1\_Block\_Grid[1]=(59,21,4), cab1\_Block\_Grid[2]=(59,22,4), cab1\_Block\_Grid[3]=(59,23,4), cab1\_Block\_Grid[4]=(59,24,4),

cab1\_Block\_Grid[5]=(59,25,4), cab1\_Block\_Grid[6]=(59,26,4), cab1\_Block\_Grid[7]=(59,27,4), cab1\_Block\_Grid[8]=(59,28,4),

cab1\_Block\_Grid[9]=(59,29,4), cab1\_Block\_Grid[10]=(59,30,4), cab1\_Block\_Grid[11]=(59,31,4), cab1\_Block\_Grid[12]=(59,32,4),

cab2\_Block\_Grid[1]=(61,21,4), cab2\_Block\_Grid[2]=(61,22,4), cab2\_Block\_Grid[3]=(61,23,4), cab2\_Block\_Grid[4]=(61,24,4),

cab2\_Block\_Grid[5]=(61,25,4), cab2\_Block\_Grid[6]=(61,26,4), cab2\_Block\_Grid[7]=(61,27,4), cab2\_Block\_Grid[8]=(61,28,4),

cab2\_Block\_Grid[9]=(61,29,4), cab2\_Block\_Grid[10]=(61,30,4), cab2\_Block\_Grid[11]=(61,31,4), cab2\_Block\_Grid[12]=(61,32,4),

cab3\_Block\_Grid[1]=(63,21,4), cab3\_Block\_Grid[2]=(63,22,4), cab3\_Block\_Grid[3]=(63,23,4), cab3\_Block\_Grid[4]=(63,24,4),

cab3\_Block\_Grid[5]=(63,25,4), cab3\_Block\_Grid[6]=(63,26,4), cab3\_Block\_Grid[7]=(63,27,4), cab3\_Block\_Grid[8]=(63,28,4),

cab3\_Block\_Grid[9]=(63,29,4), cab3\_Block\_Grid[10]=(63,30,4), cab3\_Block\_Grid[11]=(63,31,4), cab3\_Block\_Grid[12]=(63,32,4),

cab0\_All\_Blocks\_Grid=(57,20,4), cab1\_All\_Blocks\_Grid=(59,20,4), cab2\_All\_Blocks\_Grid=(61,20,4), cab3\_All\_Blocks\_Grid=(63,20,4),

Block\_Signal\_Indicator[1]=(55,21,4)

Block\_Signal\_Indicator[2]=(55,22,4)

Block\_Signal\_Indicator[3]=(55,23,4)

Block\_Signal\_Indicator[4]=(55,24,4)

Block\_Signal\_Indicator[5]=(55,25,4)

Block\_Signal\_Indicator[6]=(55,26,4)

Block\_Signal\_Indicator[7]=(55,27,4)

Block\_Signal\_Indicator[8]=(55,28,4)

Block\_Signal\_Indicator[9]=(55,29,4)

Block\_Signal\_Indicator[10]=(55,30,4)

Block\_Signal\_Indicator[11]=(55,31,4)

Block\_Signal\_Indicator[12]=(55,32,4)

Initialize\_Speed\_Index()

Initialize\_Cab\_Speed\_And\_Direction\_On\_Display()

Initialize\_Cab\_Status()

Initialize\_Cab\_Direction\_Display()

Assign\_Cab\_To\_All\_Blocks(1)

Initialize\_Detect\_Initial\_Blocks()

Initialize\_Set\_All\_Turnouts\_To\_Primary\_Direction()

Vacated\_Block\_Button = (55, 20, 4)

Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_STAGING\_LEAD] = (33, 5, 4)

Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_STAGING\_STRAIGHT] = (25, 4, 4)

Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_STAGING\_DIVERGING] = (25, 5, 4)

Extra\_Block\_Grid\_Locations[EXTRA\_BLOCK\_MINE] = (52, 11, 4)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END OF INITIALIZATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

InitStatus=Operating

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' MAIN PROGRAM

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\* Detect activation of Panic button

' If set, stop all trains and reset panel indicators

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\* Detect activation of Cab to Block assignment buttons

'\* All Block assignments

WHEN $leftmouse=cab0\_All\_Blocks\_Grid DO Assign\_Cab\_To\_All\_Blocks(0)

WHEN $leftmouse=cab1\_All\_Blocks\_Grid DO Assign\_Cab\_To\_All\_Blocks(1)

WHEN $leftmouse=cab2\_All\_Blocks\_Grid DO Assign\_Cab\_To\_All\_Blocks(2)

WHEN $leftmouse=cab3\_All\_Blocks\_Grid DO Assign\_Cab\_To\_All\_Blocks(3)

'\* Individual Block assignments

WHEN $leftmouse=cab0\_block\_grid[1] DO Assign\_Cab\_To\_Block(0,1)

WHEN $leftmouse=cab0\_block\_grid[2] DO Assign\_Cab\_To\_Block(0,2)

WHEN $leftmouse=cab0\_block\_grid[3] DO Assign\_Cab\_To\_Block(0,3)

WHEN $leftmouse=cab0\_block\_grid[4] DO Assign\_Cab\_To\_Block(0,4)

WHEN $leftmouse=cab0\_block\_grid[5] DO Assign\_Cab\_To\_Block(0,5)

WHEN $leftmouse=cab0\_block\_grid[6] DO Assign\_Cab\_To\_Block(0,6)

WHEN $leftmouse=cab0\_block\_grid[7] DO Assign\_Cab\_To\_Block(0,7)

WHEN $leftmouse=cab0\_block\_grid[8] DO Assign\_Cab\_To\_Block(0,8)

WHEN $leftmouse=cab0\_block\_grid[9] DO Assign\_Cab\_To\_Block(0,9)

WHEN $leftmouse=cab0\_block\_grid[10] DO Assign\_Cab\_To\_Block(0,10)

WHEN $leftmouse=cab0\_block\_grid[11] DO Assign\_Cab\_To\_Block(0,11)

WHEN $leftmouse=cab0\_block\_grid[12] DO Assign\_Cab\_To\_Block(0,12)

WHEN $leftmouse=cab1\_block\_grid[1] DO Assign\_Cab\_To\_Block(1,1)

WHEN $leftmouse=cab1\_block\_grid[2] DO Assign\_Cab\_To\_Block(1,2)

WHEN $leftmouse=cab1\_block\_grid[3] DO Assign\_Cab\_To\_Block(1,3)

WHEN $leftmouse=cab1\_block\_grid[4] DO Assign\_Cab\_To\_Block(1,4)

WHEN $leftmouse=cab1\_block\_grid[5] DO Assign\_Cab\_To\_Block(1,5)

WHEN $leftmouse=cab1\_block\_grid[6] DO Assign\_Cab\_To\_Block(1,6)

WHEN $leftmouse=cab1\_block\_grid[7] DO Assign\_Cab\_To\_Block(1,7)

WHEN $leftmouse=cab1\_block\_grid[8] DO Assign\_Cab\_To\_Block(1,8)

WHEN $leftmouse=cab1\_block\_grid[9] DO Assign\_Cab\_To\_Block(1,9)

WHEN $leftmouse=cab1\_block\_grid[10] DO Assign\_Cab\_To\_Block(1,10)

WHEN $leftmouse=cab1\_block\_grid[11] DO Assign\_Cab\_To\_Block(1,11)

WHEN $leftmouse=cab1\_block\_grid[12] DO Assign\_Cab\_To\_Block(1,12)

WHEN $leftmouse=cab2\_block\_grid[1] DO Assign\_Cab\_To\_Block(2,1)

WHEN $leftmouse=cab2\_block\_grid[2] DO Assign\_Cab\_To\_Block(2,2)

WHEN $leftmouse=cab2\_block\_grid[3] DO Assign\_Cab\_To\_Block(2,3)

WHEN $leftmouse=cab2\_block\_grid[4] DO Assign\_Cab\_To\_Block(2,4)

WHEN $leftmouse=cab2\_block\_grid[5] DO Assign\_Cab\_To\_Block(2,5)

WHEN $leftmouse=cab2\_block\_grid[6] DO Assign\_Cab\_To\_Block(2,6)

WHEN $leftmouse=cab2\_block\_grid[7] DO Assign\_Cab\_To\_Block(2,7)

WHEN $leftmouse=cab2\_block\_grid[8] DO Assign\_Cab\_To\_Block(2,8)

WHEN $leftmouse=cab2\_block\_grid[9] DO Assign\_Cab\_To\_Block(2,9)

WHEN $leftmouse=cab2\_block\_grid[10] DO Assign\_Cab\_To\_Block(2,10)

WHEN $leftmouse=cab2\_block\_grid[11] DO Assign\_Cab\_To\_Block(2,11)

WHEN $leftmouse=cab2\_block\_grid[12] DO Assign\_Cab\_To\_Block(2,12)

WHEN $leftmouse=cab3\_block\_grid[1] DO Assign\_Cab\_To\_Block(3,1)

WHEN $leftmouse=cab3\_block\_grid[2] DO Assign\_Cab\_To\_Block(3,2)

WHEN $leftmouse=cab3\_block\_grid[3] DO Assign\_Cab\_To\_Block(3,3)

WHEN $leftmouse=cab3\_block\_grid[4] DO Assign\_Cab\_To\_Block(3,4)

WHEN $leftmouse=cab3\_block\_grid[5] DO Assign\_Cab\_To\_Block(3,5)

WHEN $leftmouse=cab3\_block\_grid[6] DO Assign\_Cab\_To\_Block(3,6)

WHEN $leftmouse=cab3\_block\_grid[7] DO Assign\_Cab\_To\_Block(3,7)

WHEN $leftmouse=cab3\_block\_grid[8] DO Assign\_Cab\_To\_Block(3,8)

WHEN $leftmouse=cab3\_block\_grid[9] DO Assign\_Cab\_To\_Block(3,9)

WHEN $leftmouse=cab3\_block\_grid[10] DO Assign\_Cab\_To\_Block(3,10)

WHEN $leftmouse=cab3\_block\_grid[11] DO Assign\_Cab\_To\_Block(3,11)

WHEN $leftmouse=cab3\_block\_grid[12] DO Assign\_Cab\_To\_Block(3,12)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\* Detect activation of Cab Speed Buttons

When $LeftMouse = cab1\_Speed\_grid[0] DO Assign\_Speed\_To\_Cab(1, 0)

When $LeftMouse = cab1\_Speed\_grid[1] DO Assign\_Speed\_To\_Cab(1, 1)

When $LeftMouse = cab1\_Speed\_grid[2] DO Assign\_Speed\_To\_Cab(1, 2)

When $LeftMouse = cab1\_Speed\_grid[3] DO Assign\_Speed\_To\_Cab(1, 3)

When $LeftMouse = cab1\_Speed\_grid[4] DO Assign\_Speed\_To\_Cab(1, 4)

When $LeftMouse = cab1\_Speed\_grid[5] DO Assign\_Speed\_To\_Cab(1, 5)

When $LeftMouse = cab1\_Speed\_grid[6] DO Assign\_Speed\_To\_Cab(1, 6)

When $LeftMouse = cab1\_Speed\_grid[7] DO Assign\_Speed\_To\_Cab(1, 7)

When $LeftMouse = cab1\_Speed\_grid[8] DO Assign\_Speed\_To\_Cab(1, 8)

When $LeftMouse = cab1\_Speed\_grid[9] DO Assign\_Speed\_To\_Cab(1, 9)

When $LeftMouse = cab1\_Speed\_grid[10] DO Assign\_Speed\_To\_Cab(1, 10)

When $LeftMouse = cab2\_Speed\_grid[0] DO Assign\_Speed\_To\_Cab(2, 0)

When $LeftMouse = cab2\_Speed\_grid[1] DO Assign\_Speed\_To\_Cab(2, 1)

When $LeftMouse = cab2\_Speed\_grid[2] DO Assign\_Speed\_To\_Cab(2, 2)

When $LeftMouse = cab2\_Speed\_grid[3] DO Assign\_Speed\_To\_Cab(2, 3)

When $LeftMouse = cab2\_Speed\_grid[4] DO Assign\_Speed\_To\_Cab(2, 4)

When $LeftMouse = cab2\_Speed\_grid[5] DO Assign\_Speed\_To\_Cab(2, 5)

When $LeftMouse = cab2\_Speed\_grid[6] DO Assign\_Speed\_To\_Cab(2, 6)

When $LeftMouse = cab2\_Speed\_grid[7] DO Assign\_Speed\_To\_Cab(2, 7)

When $LeftMouse = cab2\_Speed\_grid[8] DO Assign\_Speed\_To\_Cab(2, 8)

When $LeftMouse = cab2\_Speed\_grid[9] DO Assign\_Speed\_To\_Cab(2, 9)

When $LeftMouse = cab2\_Speed\_grid[10] DO Assign\_Speed\_To\_Cab(2, 10)

When $LeftMouse = cab3\_Speed\_grid[0] DO Assign\_Speed\_To\_Cab(3, 0)

When $LeftMouse = cab3\_Speed\_grid[1] DO Assign\_Speed\_To\_Cab(3, 1)

When $LeftMouse = cab3\_Speed\_grid[2] DO Assign\_Speed\_To\_Cab(3, 2)

When $LeftMouse = cab3\_Speed\_grid[3] DO Assign\_Speed\_To\_Cab(3, 3)

When $LeftMouse = cab3\_Speed\_grid[4] DO Assign\_Speed\_To\_Cab(3, 4)

When $LeftMouse = cab3\_Speed\_grid[5] DO Assign\_Speed\_To\_Cab(3, 5)

When $LeftMouse = cab3\_Speed\_grid[6] DO Assign\_Speed\_To\_Cab(3, 6)

When $LeftMouse = cab3\_Speed\_grid[7] DO Assign\_Speed\_To\_Cab(3, 7)

When $LeftMouse = cab3\_Speed\_grid[8] DO Assign\_Speed\_To\_Cab(3, 8)

When $LeftMouse = cab3\_Speed\_grid[9] DO Assign\_Speed\_To\_Cab(3, 9)

When $LeftMouse = cab3\_Speed\_grid[10] DO Assign\_Speed\_To\_Cab(3, 10)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* Detect activation of pushbuttons for cab brake

'\*\*

WHEN $leftmouse = cab1\_Brake\_Grid DO \*Cab\_Brake\_Pointer[1] = \*Cab\_Brake\_Pointer[1]~

WHEN $leftmouse = cab2\_Brake\_Grid DO \*Cab\_Brake\_Pointer[2] = \*Cab\_Brake\_Pointer[2]~

WHEN $leftmouse = cab3\_Brake\_Grid DO \*Cab\_Brake\_Pointer[3] = \*Cab\_Brake\_Pointer[3]~

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* Detect activation of pushbuttons for cab direction

'\*\*

WHEN $leftmouse = cab1\_Direction\_Grid DO \*Cab\_Direction\_Pointer[1] = \*Cab\_Direction\_Pointer[1] ~

WHEN $leftmouse = cab2\_Direction\_Grid DO \*Cab\_Direction\_Pointer[2] = \*Cab\_Direction\_Pointer[2] ~

WHEN $leftmouse = cab3\_Direction\_Grid DO \*Cab\_Direction\_Pointer[3] = \*Cab\_Direction\_Pointer[3] ~

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TURNOUTS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\* Throw Tortoise Switch and corresponding display on panel

SUB Throw\_Turnout(TIndex,ctc\_Grid)

$Switch(ctc\_Grid)=$Switch(ctc\_Grid)~ 'Throw turnout on panel display

\*Turnout\_Pointer[TIndex]=\*Turnout\_Pointer[TIndex]~, 'Reverse power to Tortoise & LED or to Atlas LED only

Turnout\_Status[TIndex]=Turnout\_Status[TIndex]~, 'Change direction status of turnout

IF Turnout\_Type[TIndex]=ATLAS THEN

IF Turnout\_Status[TIndex]=TURNOUT\_DIRECTION\_PRIMARY THEN 'Determine new status of Atlas turnout

\*Atlas\_Turnout\_Pointer\_Secondary[TIndex]=PULSE 0.25, ' Pulse power to proper Atlas leg

\*Atlas\_Turnout\_LED\_Pointer\_Secondary[TIndex]=TURNOUT\_DIRECTION\_SECONDARY 'NOT USED?

ELSE \*Atlas\_Turnout\_Pointer\_Primary[TIndex]=PULSE 0.25,

\*Atlas\_Turnout\_led\_Pointer\_Primary[TIndex]=TURNOUT\_DIRECTION\_PRIMARY, 'NOT USED?

ENDIF

ENDIF

ENDSUB

'\*\* Detect activation of turnouts on panel or Turnout Micro Switches

' Call subroutine

' Left click changes tortoise direction, indicating diverging route with red on panel, main route with green

WHEN $Leftmouse=Turnout\_Grid[1] or \*Turnout\_Button\_Pointer[1]=on DO Throw\_Turnout(1,Turnout\_Grid[1])

WHEN $Leftmouse=Turnout\_Grid[2] or \*Turnout\_Button\_Pointer[2]=on DO Throw\_Turnout(2,Turnout\_Grid[2])

WHEN $Leftmouse=Turnout\_Grid[3] or \*Turnout\_Button\_Pointer[3]=on DO Throw\_Turnout(3,Turnout\_Grid[3]), Redraw\_Extra\_Blocks\_On\_Grid()

WHEN $Leftmouse=Turnout\_Grid[4] or \*Turnout\_Button\_Pointer[4]=on DO Throw\_Turnout(4,Turnout\_Grid[4])

WHEN $Leftmouse=Turnout\_Grid[5] or \*Turnout\_Button\_Pointer[5]=on DO Throw\_Turnout(5,Turnout\_Grid[5])

WHEN $Leftmouse=Turnout\_Grid[6] or \*Turnout\_Button\_Pointer[6]=on DO Throw\_Turnout(6,Turnout\_Grid[6])

WHEN $Leftmouse=Turnout\_Grid[7] or \*Turnout\_Button\_Pointer[7]=on DO Throw\_Turnout(7,Turnout\_Grid[7])

WHEN $Leftmouse=Turnout\_Grid[8] or \*Turnout\_Button\_Pointer[8]=on DO Throw\_Turnout(8,Turnout\_Grid[8])

WHEN $Leftmouse=Turnout\_Grid[9] or \*Turnout\_Button\_Pointer[9]=on DO Throw\_Turnout(9,Turnout\_Grid[9])

WHEN $Leftmouse=Turnout\_Grid[10] or \*Turnout\_Button\_Pointer[10]=on DO Throw\_Turnout(10,Turnout\_Grid[10])

WHEN $Leftmouse=Turnout\_Grid[11] or \*Turnout\_Button\_Pointer[11]=on DO Throw\_Turnout(11,Turnout\_Grid[11])

WHEN $Leftmouse=Turnout\_Grid[12] or \*Turnout\_Button\_Pointer[12]=on DO Throw\_Turnout(12,Turnout\_Grid[12]), Redraw\_Extra\_Blocks\_On\_Grid()

WHEN $Leftmouse=Turnout\_Grid[13] or \*Turnout\_Button\_Pointer[13]=on DO Throw\_Turnout(13,Turnout\_Grid[13])

WHEN $Leftmouse=Turnout\_Grid[14] or \*Turnout\_Button\_Pointer[14]=on DO Throw\_Turnout(14,Turnout\_Grid[14])

WHEN $Leftmouse=Turnout\_Grid[15] or \*Turnout\_Button\_Pointer[15]=on DO Throw\_Turnout(15,Turnout\_Grid[15])

WHEN $Leftmouse=Turnout\_Grid[16] or \*Turnout\_Button\_Pointer[16]=on DO Throw\_Turnout(16,Turnout\_Grid[16])

WHEN $Leftmouse=Turnout\_Grid[17] or \*Turnout\_Button\_Pointer[17]=on DO Throw\_Turnout(17,Turnout\_Grid[17])

WHEN $Leftmouse=Turnout\_Grid[18] or \*Turnout\_Button\_Pointer[18]=on DO Throw\_Turnout(18,Turnout\_Grid[18]), Redraw\_Extra\_Blocks\_On\_Grid()

WHEN $Leftmouse=Turnout\_Grid[19] or \*Turnout\_Button\_Pointer[19]=on DO Throw\_Turnout(19,Turnout\_Grid[19])

WHEN $Leftmouse=Turnout\_Grid[20] or \*Turnout\_Button\_Pointer[20]=on DO Throw\_Turnout(20,Turnout\_Grid[20])

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\* LOOP through East and West current detectors to determine block occupancy

WHEN \*CD\_East\_Pointer[1] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_East(1)

WHEN \*CD\_West\_Pointer[1] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_West(1)

WHEN \*CD\_East\_Pointer[1] = DETECTOR\_BLOCK\_VACANT, \*CD\_West\_Pointer[1] = DETECTOR\_BLOCK\_VACANT DO Current\_Detector\_Stopped\_Triggering(1)

WHEN Block\_Status[1] = BLOCK\_STATUS\_VACANT DO Release\_Hold\_On\_Block(1)

WHEN \*CD\_East\_Pointer[2] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_East(2)

WHEN \*CD\_West\_Pointer[2] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_West(2)

WHEN \*CD\_East\_Pointer[2] = DETECTOR\_BLOCK\_VACANT, \*CD\_West\_Pointer[2] = DETECTOR\_BLOCK\_VACANT DO Current\_Detector\_Stopped\_Triggering(2)

WHEN Block\_Status[2] = BLOCK\_STATUS\_VACANT DO Release\_Hold\_On\_Block(2)

WHEN \*CD\_East\_Pointer[3] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_East(3)

WHEN \*CD\_West\_Pointer[3] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_West(3)

WHEN \*CD\_East\_Pointer[3] = DETECTOR\_BLOCK\_VACANT, \*CD\_West\_Pointer[3] = DETECTOR\_BLOCK\_VACANT DO Current\_Detector\_Stopped\_Triggering(3)

WHEN Block\_Status[3] = BLOCK\_STATUS\_VACANT DO Release\_Hold\_On\_Block(3)

WHEN \*CD\_East\_Pointer[4] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_East(4)

WHEN \*CD\_West\_Pointer[4] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_West(4)

WHEN \*CD\_East\_Pointer[4] = DETECTOR\_BLOCK\_VACANT, \*CD\_West\_Pointer[4] = DETECTOR\_BLOCK\_VACANT DO Current\_Detector\_Stopped\_Triggering(4)

WHEN Block\_Status[4] = BLOCK\_STATUS\_VACANT DO Release\_Hold\_On\_Block(4)

WHEN \*CD\_East\_Pointer[5] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_East(5)

WHEN \*CD\_West\_Pointer[5] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_West(5)

WHEN \*CD\_East\_Pointer[5] = DETECTOR\_BLOCK\_VACANT, \*CD\_West\_Pointer[5] = DETECTOR\_BLOCK\_VACANT DO Current\_Detector\_Stopped\_Triggering(5)

WHEN Block\_Status[5] = BLOCK\_STATUS\_VACANT DO Release\_Hold\_On\_Block(5)

WHEN \*CD\_East\_Pointer[6] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_East(6)

WHEN \*CD\_West\_Pointer[6] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_West(6)

WHEN \*CD\_East\_Pointer[6] = DETECTOR\_BLOCK\_VACANT, \*CD\_West\_Pointer[6] = DETECTOR\_BLOCK\_VACANT DO Current\_Detector\_Stopped\_Triggering(6)

WHEN Block\_Status[6] = BLOCK\_STATUS\_VACANT DO Release\_Hold\_On\_Block(6)

WHEN \*CD\_East\_Pointer[7] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_East(7)

WHEN \*CD\_West\_Pointer[7] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_West(7)

WHEN \*CD\_East\_Pointer[7] = DETECTOR\_BLOCK\_VACANT, \*CD\_West\_Pointer[7] = DETECTOR\_BLOCK\_VACANT DO Current\_Detector\_Stopped\_Triggering(7)

WHEN Block\_Status[7] = BLOCK\_STATUS\_VACANT DO Release\_Hold\_On\_Block(7)

WHEN \*CD\_East\_Pointer[8] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_East(8)

WHEN \*CD\_West\_Pointer[8] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_West(8)

WHEN \*CD\_East\_Pointer[8] = DETECTOR\_BLOCK\_VACANT, \*CD\_West\_Pointer[8] = DETECTOR\_BLOCK\_VACANT DO Current\_Detector\_Stopped\_Triggering(8)

WHEN Block\_Status[8] = BLOCK\_STATUS\_VACANT DO Release\_Hold\_On\_Block(8)

WHEN \*CD\_East\_Pointer[9] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_East(9)

WHEN \*CD\_West\_Pointer[9] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_West(9)

WHEN \*CD\_East\_Pointer[9] = DETECTOR\_BLOCK\_VACANT, \*CD\_West\_Pointer[9] = DETECTOR\_BLOCK\_VACANT DO Current\_Detector\_Stopped\_Triggering(9)

WHEN Block\_Status[9] = BLOCK\_STATUS\_VACANT DO Release\_Hold\_On\_Block(9)

WHEN \*CD\_East\_Pointer[10] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_East(10)

WHEN \*CD\_West\_Pointer[10] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_West(10)

WHEN \*CD\_East\_Pointer[10] = DETECTOR\_BLOCK\_VACANT, \*CD\_West\_Pointer[10] = DETECTOR\_BLOCK\_VACANT DO Current\_Detector\_Stopped\_Triggering(10)

WHEN Block\_Status[10] = BLOCK\_STATUS\_VACANT DO Release\_Hold\_On\_Block(10)

WHEN \*CD\_East\_Pointer[11] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_East(11)

WHEN \*CD\_West\_Pointer[11] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_West(11)

WHEN \*CD\_East\_Pointer[11] = DETECTOR\_BLOCK\_VACANT, \*CD\_West\_Pointer[11] = DETECTOR\_BLOCK\_VACANT DO Current\_Detector\_Stopped\_Triggering(11)

WHEN Block\_Status[11] = BLOCK\_STATUS\_VACANT DO Release\_Hold\_On\_Block(11)

WHEN \*CD\_East\_Pointer[12] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_East(12)

WHEN \*CD\_West\_Pointer[12] = DETECTOR\_BLOCK\_OCCUPIED DO Current\_Detector\_Triggered\_West(12)

WHEN \*CD\_East\_Pointer[12] = DETECTOR\_BLOCK\_VACANT, \*CD\_West\_Pointer[12] = DETECTOR\_BLOCK\_VACANT DO Current\_Detector\_Stopped\_Triggering(12)

WHEN Block\_Status[12] = BLOCK\_STATUS\_VACANT DO Release\_Hold\_On\_Block(12)

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' LOOP through East and West InfraRed detectors to determine block occupancy

' IF \*IRD\_East\_Pointer[OccupancyIndex]=DETECTOR\_BLOCK\_OCCUPIED THEN

' $Draw Sprite(IR\_Detector\_Grid\_E[OccupancyIndex])= IR\_Detector\_Shape\_E[OccupancyIndex] in red

' ELSE $Erase Sprite(IR\_Detector\_Grid\_E[OccupancyIndex])

' ENDIF

'

' IF \*IRD\_West\_Pointer[OccupancyIndex]=DETECTOR\_BLOCK\_OCCUPIED THEN

' $Draw Sprite(IR\_Detector\_Grid\_W[OccupancyIndex])= IR\_Detector\_Shape\_W[OccupancyIndex] in red

' ELSE $Erase Sprite(IR\_Detector\_Grid\_W[OccupancyIndex])

' ENDIF

'

' IF OccupyEast[OccupancyIndex]=On or

' OccupyWest[OccupancyIndex] = on or

' \*IRD\_East\_Pointer[OccupancyIndex]=on or

' \*IRD\_West\_pointer[OccupancyIndex]= on

' THEN

' Block\_Status[OccupancyIndex]=1,

' ELSE Block\_Status[OccupancyIndex]=0

' ENDIF

'

' ALWAYS DO

' IF OccupancyIndex<MAX\_BLOCK then OccupancyIndex=+

'

' ENDIF

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' LOOP through EOT InfraRed detectors to determine block occupancy

' WHEN IR\_EOT\_Detector[0]=DETECTOR\_BLOCK\_OCCUPIED DO

' $Draw Sprite(IR\_EOT\_Detector\_Grid[0])= IR\_EOT\_Detector\_Shape[0] in red

' WHEN IR\_EOT\_Detector[0]=DETECTOR\_BLOCK\_VACANT DO

' $Erase Sprite(IR\_EOT\_Detector\_Grid[0])

' \*\*\*\*\*\*\*\*\*\*\*\*\*\* DCC CABS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SUB Update\_Dcc\_Cab\_Speed(DccCabIndex, {Local} NewSpeed)

NewSpeed = \*Dcc\_Cab\_Pointer[DccCabIndex].Speed

Previous\_Dcc\_Cab\_Speed[DccCabIndex] = \*Dcc\_Cab\_Pointer[DccCabIndex].Speed

IF NewSpeed > 100 THEN NewSpeed = 100 ENDIF

\*Cab\_Pointer[DccCabIndex].Speed = NewSpeed

ENDSUB

SUB Update\_Cab\_Speed(CabIndex)

\*Dcc\_Cab\_Pointer[CabIndex].Speed = \*Cab\_Pointer[CabIndex].Speed

Previous\_Cab\_Speed[CabIndex] = \*Cab\_Pointer[CabIndex].Speed

ENDSUB

WHEN \*Dcc\_Cab\_Pointer[1].Speed <> Previous\_Dcc\_Cab\_Speed[1] DO Update\_Dcc\_Cab\_Speed(1)

WHEN \*Dcc\_Cab\_Pointer[2].Speed <> Previous\_Dcc\_Cab\_Speed[2] DO Update\_Dcc\_Cab\_Speed(2)

WHEN \*Dcc\_Cab\_Pointer[3].Speed <> Previous\_Dcc\_Cab\_Speed[3] DO Update\_Dcc\_Cab\_Speed(3)

WHEN \*Dcc\_Cab\_Pointer[1].Brake = ON DO \*Cab\_Pointer[1].Brake = ON, \*Dcc\_Cab\_Pointer[1].F1 = ON

WHEN \*Dcc\_Cab\_Pointer[1].Brake = OFF DO \*Cab\_Pointer[1].Brake = OFF, \*Dcc\_Cab\_Pointer[1].F1 = OFF

WHEN \*Dcc\_Cab\_Pointer[2].Brake = ON DO \*Cab\_Pointer[2].Brake = ON, \*Dcc\_Cab\_Pointer[2].F1 = ON

WHEN \*Dcc\_Cab\_Pointer[2].Brake = OFF DO \*Cab\_Pointer[2].Brake = OFF, \*Dcc\_Cab\_Pointer[2].F1 = OFF

WHEN \*Dcc\_Cab\_Pointer[3].Brake = ON DO \*Cab\_Pointer[3].Brake = ON, \*Dcc\_Cab\_Pointer[3].F1 = ON

WHEN \*Dcc\_Cab\_Pointer[4].Brake = OFF DO \*Cab\_Pointer[3].Brake = OFF, \*Dcc\_Cab\_Pointer[3].F1 = OFF

WHEN \*Dcc\_Cab\_Pointer[1].F1 = ON DO \*Dcc\_Cab\_Pointer[1].Brake = ON

WHEN \*Dcc\_Cab\_Pointer[1].F1 = OFF DO \*Dcc\_Cab\_Pointer[1].Brake = OFF

WHEN \*Dcc\_Cab\_Pointer[2].F1 = ON DO \*Dcc\_Cab\_Pointer[2].Brake = ON

WHEN \*Dcc\_Cab\_Pointer[2].F1 = OFF DO \*Dcc\_Cab\_Pointer[2].Brake = OFF

WHEN \*Dcc\_Cab\_Pointer[3].F1 = ON DO \*Dcc\_Cab\_Pointer[3].Brake = ON

WHEN \*Dcc\_Cab\_Pointer[3].F1 = OFF DO \*Dcc\_Cab\_Pointer[4].Brake = OFF

WHEN \*Dcc\_Cab\_Pointer[1].Direction = FORWARD DO \*Cab\_Pointer[1].Direction = FORWARD

WHEN \*Dcc\_Cab\_Pointer[1].Direction = REVERSE DO \*Cab\_Pointer[1].Direction = REVERSE

WHEN \*Dcc\_Cab\_Pointer[2].Direction = FORWARD DO \*Cab\_Pointer[2].Direction = FORWARD

WHEN \*Dcc\_Cab\_Pointer[2].Direction = REVERSE DO \*Cab\_Pointer[2].Direction = REVERSE

WHEN \*Dcc\_Cab\_Pointer[3].Direction = FORWARD DO \*Cab\_Pointer[3].Direction = FORWARD

WHEN \*Dcc\_Cab\_Pointer[3].Direction = REVERSE DO \*Cab\_Pointer[3].Direction = REVERSE

' \*\*\* --- \*\*\*

WHEN \*Cab\_Pointer[1].Speed <> Previous\_Cab\_Speed[1] DO Update\_Cab\_Speed(1)

WHEN \*Cab\_Pointer[2].Speed <> Previous\_Cab\_Speed[2] DO Update\_Cab\_Speed(2)

WHEN \*Cab\_Pointer[3].Speed <> Previous\_Cab\_Speed[3] DO Update\_Cab\_Speed(3)

WHEN \*Cab\_Pointer[1].Brake = ON DO \*Dcc\_Cab\_Pointer[1].Brake = ON

WHEN \*Cab\_Pointer[1].Brake = OFF DO \*Dcc\_Cab\_Pointer[1].Brake = OFF

WHEN \*Cab\_Pointer[2].Brake = ON DO \*Dcc\_Cab\_Pointer[2].Brake = ON

WHEN \*Cab\_Pointer[2].Brake = OFF DO \*Dcc\_Cab\_Pointer[2].Brake = OFF

WHEN \*Cab\_Pointer[3].Brake = ON DO \*Dcc\_Cab\_Pointer[3].Brake = ON

WHEN \*Cab\_Pointer[4].Brake = OFF DO \*Dcc\_Cab\_Pointer[3].Brake = OFF

WHEN \*Cab\_Pointer[1].Direction = FORWARD DO \*Dcc\_Cab\_Pointer[1].Direction = FORWARD

WHEN \*Cab\_Pointer[1].Direction = REVERSE DO \*Dcc\_Cab\_Pointer[1].Direction = REVERSE

WHEN \*Cab\_Pointer[2].Direction = FORWARD DO \*Dcc\_Cab\_Pointer[2].Direction = FORWARD

WHEN \*Cab\_Pointer[2].Direction = REVERSE DO \*Dcc\_Cab\_Pointer[2].Direction = REVERSE

WHEN \*Cab\_Pointer[3].Direction = FORWARD DO \*Dcc\_Cab\_Pointer[3].Direction = FORWARD

WHEN \*Cab\_Pointer[3].Direction = REVERSE DO \*Dcc\_Cab\_Pointer[3].Direction = REVERSE

'

'

SUB Redraw\_Cab\_Speed(CabIndex, {Local} Index, DrawColor)

Index = 0

UNTIL Index >= MAX\_SPEED\_NOTCH\_INDEX LOOP

IF \*Cab\_Pointer[CabIndex].Speed >= Speed\_Notch[Index] THEN

DrawColor = Cab\_Color[CabIndex]

ELSE

DrawColor = COLOR\_UNSELECTED\_SPEED

ENDIF

IF CabIndex = 1 THEN

$Color Track(cab1\_Speed\_Grid[Index]) = DrawColor

ELSEIF CabIndex = 2 THEN

$Color Track(cab2\_Speed\_Grid[Index]) = DrawColor

ELSEIF CabIndex = 3 THEN

$Color Track(cab3\_Speed\_Grid[Index]) = DrawColor

ENDIF

Index = 1 +

ENDLOOP

ENDSUB

SUB Redraw\_Cab\_Brake(CabIndex, {Local} DrawColor)

IF \*Cab\_Pointer[CabIndex].Brake = ON THEN

DrawColor = COLOR\_BRAKE\_ON

ELSE

DrawColor = COLOR\_BRAKE\_OFF

ENDIF

IF CabIndex = 1 THEN

$Color Track(cab1\_Brake\_Grid) = DrawColor

ELSEIF CabIndex = 2 THEN

$Color Track(cab2\_Brake\_Grid) = DrawColor

ELSEIF CabIndex = 3 THEN

$Color Track(cab3\_Brake\_Grid) = DrawColor

ENDIF

ENDSUB

SUB Redraw\_Cab\_Direction(CabIndex, {Local} DrawColor, DrawIcon)

IF \*Cab\_Pointer[CabIndex].Direction = FORWARD THEN

DrawColor = COLOR\_DIRECTION\_FORWARD

DrawIcon = ICON\_DIRECTION\_FORWARD

ELSE

DrawColor = COLOR\_DIRECTION\_REVERSE

DrawIcon = ICON\_DIRECTION\_REVERSE

ENDIF

IF CabIndex = 1 THEN

$Draw Sprite(Cab1\_F\_R\_Indication\_Grid) = DrawIcon in DrawColor

ELSEIF CabIndex = 2 THEN

$Draw Sprite(Cab2\_F\_R\_Indication\_Grid) = DrawIcon in DrawColor

ELSEIF CabIndex = 3 THEN

$Draw Sprite(Cab3\_F\_R\_Indication\_Grid) = DrawIcon in DrawColor

ENDIF

ENDSUB

WHEN \*Cab\_Pointer[1].Speed <> Previous\_Cab\_Speed[1] DO Redraw\_Cab\_Speed(1)

WHEN \*Cab\_Pointer[2].Speed <> Previous\_Cab\_Speed[2] DO Redraw\_Cab\_Speed(2)

WHEN \*Cab\_Pointer[3].Speed <> Previous\_Cab\_Speed[3] DO Redraw\_Cab\_Speed(3)

WHEN \*Cab\_Pointer[1].Brake = ON DO Redraw\_Cab\_Brake(1)

WHEN \*Cab\_Pointer[1].Brake = OFF DO Redraw\_Cab\_Brake(1)

WHEN \*Cab\_Pointer[2].Brake = ON DO Redraw\_Cab\_Brake(2)

WHEN \*Cab\_Pointer[2].Brake = OFF DO Redraw\_Cab\_Brake(2)

WHEN \*Cab\_Pointer[3].Brake = ON DO Redraw\_Cab\_Brake(3)

WHEN \*Cab\_Pointer[3].Brake = OFF DO Redraw\_Cab\_Brake(3)

WHEN \*Cab\_Pointer[1].Direction = FORWARD DO Redraw\_Cab\_Direction(1)

WHEN \*Cab\_Pointer[1].Direction = REVERSE DO Redraw\_Cab\_Direction(1)

WHEN \*Cab\_Pointer[2].Direction = FORWARD DO Redraw\_Cab\_Direction(2)

WHEN \*Cab\_Pointer[2].Direction = REVERSE DO Redraw\_Cab\_Direction(2)

WHEN \*Cab\_Pointer[3].Direction = FORWARD DO Redraw\_Cab\_Direction(3)

WHEN \*Cab\_Pointer[3].Direction = REVERSE DO Redraw\_Cab\_Direction(3)

''''''''''''''''''''''''

'' Manually Calls on Blocks

''''''''''''''''''''''''

SUB Call\_For\_Manual\_Block(BlockIndex)

IF Block\_Status[BlockIndex] >= BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA THEN

Block\_Status[BlockIndex] = BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA -

ELSE

Block\_Status[BlockIndex] = BLOCK\_STATUS\_MANUAL\_HOLD\_DELTA +

ENDIF

Redraw\_Cab\_Block\_All(BlockIndex)

ENDSUB

WHEN $LeftMouse = Block\_Signal\_Indicator[1] DO Call\_For\_Manual\_Block(1)

WHEN $LeftMouse = Block\_Signal\_Indicator[2] DO Call\_For\_Manual\_Block(2)

WHEN $LeftMouse = Block\_Signal\_Indicator[3] DO Call\_For\_Manual\_Block(3)

WHEN $LeftMouse = Block\_Signal\_Indicator[4] DO Call\_For\_Manual\_Block(4)

WHEN $LeftMouse = Block\_Signal\_Indicator[5] DO Call\_For\_Manual\_Block(5)

WHEN $LeftMouse = Block\_Signal\_Indicator[6] DO Call\_For\_Manual\_Block(6)

WHEN $LeftMouse = Block\_Signal\_Indicator[7] DO Call\_For\_Manual\_Block(7)

WHEN $LeftMouse = Block\_Signal\_Indicator[8] DO Call\_For\_Manual\_Block(8)

WHEN $LeftMouse = Block\_Signal\_Indicator[9] DO Call\_For\_Manual\_Block(9)

WHEN $LeftMouse = Block\_Signal\_Indicator[10] DO Call\_For\_Manual\_Block(10)

WHEN $LeftMouse = Block\_Signal\_Indicator[11] DO Call\_For\_Manual\_Block(11)

WHEN $LeftMouse = Block\_Signal\_Indicator[12] DO Call\_For\_Manual\_Block(12)

''

'' VACATED BLOCK

''

WHEN $Leftmouse = Vacated\_Block\_Button DO LAYOUT\_CONTROL\_INCLUDE\_VACATED\_BLOCK = LAYOUT\_CONTROL\_INCLUDE\_VACATED\_BLOCK ~

WHEN LAYOUT\_CONTROL\_INCLUDE\_VACATED\_BLOCK = True DO $Color Track(Vacated\_Block\_Button) = COLOR\_INCLUDE\_VACATED\_BLOCK

WHEN LAYOUT\_CONTROL\_INCLUDE\_VACATED\_BLOCK = False DO $Color Track(Vacated\_Block\_Button) = COLOR\_DONT\_INCLUDE\_VACATED\_BLOCK