This document is to conceptualize our process

STEPS

~~STEP 1 DONE~~
~~[test]<----->[requirements]<---~~**~~WE NEED TO LINK THESE 2--~~** ~~>[flow chart model]~~

STEP 2

- Edit input signals to test flow chart transitions
- For Basic Case. A Couple Input signals for each test case
    - For change2park, make it go to neutral, then go into park
    - Maybe even do park -> neutral -> reverse(reversing out of parkway) -> drive(driving on road) -> reverse(reverse into parking spot) -> park.
    - Create like 2 for each maybe
- For Advanced case. Create a ‘golden scenario’, that shows how the car should work if driver selects multiple gears. Will be useful when we break the system.
    - Input variables are scenarios we are making:
        - Exp: parked -> reversing out -> driving -> flat tire(neutral) -> park at gas station -> drive home -> reverse into parking spot -> park

STEP 2.1

- Create a list of conditions to switch into a certain gear on the google doc, it will be easier to create our scenario in signal editor if we do this. For exp. To get into drive var1 == 1, var 2 == 0, var4 != 4, etc.

STEP 3

- Logical and Temporal Assessments
    - We are Creating constraints/conditions that will cross reference the output, and tell us where it went wrong, if it did
    - We need to make these based on the arrow conditions in the flow chart, as well as anything else we can think of(prolly not too many)
- Logical Assessments are checking if value x is true when value y == 1 or something. Check if 2 variables are working correctly together.
- Temporal Assessments are checking if event A occurs then event B must happen within t secs, or signal C is high for this many seconds.
- Makes it so we dont have to eyeball these output signals

STEP 4: ADVANCED SUBMISSION

- Do 5 logical assessments, showing the output move from FALSE to TRUE with the chane of 1 input variable at time t. 2/3 conditions met, output = FALSE, then 3/3 conditions met, output = TRUE. Do this for 5 req cases.

What Variables do

INPUTS
ElectronicTransmissionRangeSel - the range(gear) selector
- Only used when shifting in and out of park

ElectricParkBrakeAvailable
ElectricParkBrakeApplied
ParktoChargeRequested - car requested it needs to charge, another pre-req to go into park
Park_BrakePressed - is brake applied or nah, cant change into or outta park w/o this == 1
VehicleSpeed
HVShutdownRequested - High Voltage Shutdown, condition to force it to go into neutral or park
VehicleSystemEnabled - is the propulsion system on, can't go into reverse or drive w/o this

OUTPUTS
TransmissionPosition - final executed transmission range, what the rest of the vehicle sees atp. What is the real position of the transmission.
TransmissionShiftPosition - driver's last input, where the gear shift it atp
TransmissionEstimatedGear - car predicting what gear the car could go into next, determined by some external logic. Is the same as transmissionGearSelected in this project
TransmissionGearSelected - what the controller, having taken the driver input and estimated gear logic into consideration, has chosen to command the transmission to do.
TranmissionEngagedState - is the transmission engaged(torque locked in or not) (0=disengaged, 1=forward engaged, 2=reverse engaged)
TranmissionEngagedDirectionState - what the direction of the gear will be based on the engaged state (1 = Neutral/Park (no torque), 2 = Forward, 3 = Reverse)
ElectronicTransmissionRangeState - what state the finite state machine is currently in (0 = Park, 1 = Neutral, 2 = Drive, 3 = Reverse, 4 = Neutral_Intermediate_Drive, 5 = Neutral_Intermediate_Reverse.)

HAO'S NOTES
Electronic Transmission Range Selection (ERTS) – the range selector
Ranges are what the driver selects: [PARK, REVERSE, NEUTRAL and DRIVE]
Only used when shifting in and out of park

Drive to Park Process:
Driver will send range change request to Park
System will set Transmission Position to Neutral Position
Parking brake will physically engage
System will set Transmission Position to Parking Position

Park to Drive Process:
Driver will send range change request to Drive

System will set Transmission Position to Neutral Position
Parking brake will physically disengage
System will set Transmission Position to Drive Position / Reverse Position

Pseudo-Code

Requirement 1
If the following:
Electronic Transmission Range Primary – PARK
Charging System Shift to Park Request – TRUE
EPB Application Status – TRUE
EPB Availability Status – AVAILABLE
Vehicle Speed < 0.5
Then:
    Transmission Actual Range – PARK

Keep Transmission Actual Range == PARK if
    Charging System Shift to Park Request – TRUE
    Transmission Actual Range – PARK

1.2

If "ElectronicTransmissionRangeSel" indicates REVERSE and "ElectricParkBrakeApplied" is False and the Propulsion system is Active, "TransmissionPosition" shall indicate REVERSE

// to make sure that etrs = rev has the intended effect and moves the transmission to rev

POSSIBLE TESTS / INPUT SIGNALS
1.) test ElectronicTransmissionRangeSel = REVERSE all the way, ElectricParkBrakeApplied == 2 all the way, VehicleSystemEnabled == 1 all the way
- success state

3.) test ElectronicTransmissionRangeSel = drive all the way, ElectricParkBrakeApplied == 2 all the way, VehicleSystemEnabled == 1 all the way
- expecting fail state (stuck in drive)
4.) test ElectronicTransmissionRangeSel = REVERSE all the way, ElectricParkBrakeApplied == 1 all the way, VehicleSystemEnabled == 1 all the way
- expecting fail state (stuck in park)

5.) test ElectronicTransmissionRangeSel = drive 0<t<10, ElectronicTransmissionRangeSel = REVERSE 10<t<35, ElectricParkBrakeApplied == 1 all the way, VehicleSystemEnabled == 1 all the way
- temporal test, expecting success state for t>10

OTHER INPUT VARIABLES
ElectricParkBrakeAvailable
ElectricParkBrakeAvailable == 1
doesn't matter really in this case, but prolly want that in general ;D
ParktoChargeRequested
ParktoChargeRequested == 0
We want to go to reverse, not go to park
Park_BrakePressed
Park_BrakePressed == 1
to get out of park
VehicleSpeed
VehicleSpeed < 0.5
VehicleSpeed == 0.1

to get out of park

HVShutdownRequested - High Voltage Shutdown, condition to force it to go into neutral

HVShutdownRequested == 0

We dont want to shut down power, we need power to go in REVERSE

1.3

If "ElectronicTransmissionRangeSel" indicates NEUTRAL, "TransmissionPosition" shall indicate NEUTRAL

// to make sure that etrs = neutral has the intended effect and moves the transmission to neutral

POSSIBLE TESTS / INPUT SIGNALS
1.) test ElectronicTransmissionRangeSel = NEUTRAL all the way
- success state

3.) test ElectronicTransmissionRangeSel = drive all the way
- expecting fail state (stuck in drive)
4.) test ElectronicTransmissionRangeSel = rev all the way
- expecting fail state (stuck in rev)

5.) test ElectronicTransmissionRangeSel = rev 0<t<10, ElectronicTransmissionRangeSel = NEUTRAL 10<t<35
- temporal test, expecting success state for t>10

OTHER INPUT VARIABLES
ElectricParkBrakeApplied
ElectricParkBrakeApplied == 2
ElectricParkBrakeAvailable
ElectricParkBrakeAvailable == 1
doesn't matter really in this case, but prolly want that in general ;D
ParktoChargeRequested
ParktoChargeRequested == 0
We want to stay in neutral, not go to park
Park_BrakePressed
Park_BrakePressed == 1
VehicleSpeed
VehicleSpeed < 0.5
VehicleSpeed == 0.1
HVShutdownRequested - High Voltage Shutdown, condition to force it to go into neutral
HVShutdownRequested == 0
We want ElectronicTransmissionRangeSel to control the car going into neutral, not the other variables, like HVShutdownRequested
VehicleSystemEnabled - is the propulsion system on, can't go into reverse or drive w/o this, basically shutting off the engine
VehicleSystemEnabled == 1
Doesn't rlly matter in this case

1.4

If "ElectronicTransmissionRangeSel" indicates DRIVE and "ElectricParkBrakeApplied" is False AND the Propulsion system is Active, "TransmissionPosition" shall indicate DRIVE

//just testing if it can go into drive

POSSIBLE TESTS / INPUT SIGNALS

1.) test ElectronicTransmissionRangeSel = drive all the way && ElectricParkBrakeApplied = false all the way && VehicleSystemEnabled == 1 all the way

- success state

3.) test ElectronicTransmissionRangeSel = drive all the way && ElectricParkBrakeApplied = false all the way && VehicleSystemEnabled == 0 all the way

- expecting fail state (stuck in neutral)

4.) test ElectronicTransmissionRangeSel = drive all the way && ElectricParkBrakeApplied = True all the way && VehicleSystemEnabled == 1 all the way

- expecting fail state (stuck in neutral)

5.) test ElectronicTransmissionRangeSel = rev all the way && ElectricParkBrakeApplied = false all the way && VehicleSystemEnabled == 1 all the way

- expecting fail state (stuck in rev)

6.) test ElectronicTransmissionRangeSel = drive all the way && ElectricParkBrakeApplied = True 0<t<10, ElectricParkBrakeApplied = False 10<t<35. && VehicleSystemEnabled == 1 all the way

- temporal test, expecting succes state for t>10

7.) test ElectronicTransmissionRangeSel = rev 0<t<10, ElectronicTransmissionRangeSel = drive 10<t<35 && ElectricParkBrakeApplied all the way.&& VehicleSystemEnabled == 1 all the way

- temporal test, expecitng succes state for t>10

OTHER INPUT VARIABLES

ElectricParkBrakeAvailable

ElectricParkBrakeAvailable == 1

doesn't matter really in this case, but prolly want that in general ;D

ParktoChargeRequested

ParktoChargeRequested == 0

Since we are going to drive we don't want this at all.

Park_BrakePressed

Park_BrakePressed == 1

Just in case another transition on the way to drive needs it to be activated

VehicleSpeed

VehicleSpeed < 0.5

VehicleSpeed == 0.1

Just in case another transition on the way to drive needs it to be < 0.5

HVShutdownRequested - High Voltage Shutdown, condition to force it to go into neutral or park

HVShutdownRequested == 0

Since we are going to drive we don't want this at all.

2.1

If "ElectronicTransmissionRangeSel" indicates DRIVE/REVERSE (2 or 4) and the ElectricParkBrakeApplied is False(2), the PSC shall indicate NEUTRAL via "TransmissionPosition"

// this case does not work unless the

1.) propulsion system is turned off, VehicleSystemEnabled ~= 1
2.) or we quickly turn the ElectricParkBrakeApplied == True(1) after setting it == False(2) for a quick moment,
    a.) to get it out of park, towards drive; we set ElectricParkBrakeApplied == 1 right after it gets into neutral to prevent it from moving to drive/rev

```
/*
PARK     NEUT     DRIVE
[ ]<-----> [ ] <-----> [ ]
            ^
            |
            |
            v
           [ ]
           REVERSE
*/
```

NEUT --> REV

```
[ElectronicTransmissionRangeSel == 2  && ElectricParkBrakeApplied == 2 && VehicleSystemEnabled == 1]
```

NEUT --> DRIVE

```
[ElectronicTransmissionRangeSel == 4 && ElectricParkBrakeApplied == 2 && VehicleSystemEnabled == 1]
```

PARK --> NEUT

```
[((ElectronicTransmissionRangeSel ~= 1 && ElectronicTransmissionRangeSel ~= 0) || (ElectricParkBrakeApplied == 2 || ElectricParkBrakeApplied == 4))...
&& ParktoChargeRequested == 0 && VehicleSpeed < 0.5 && Park_BrakePressed == 1]
```

```
/*
In all these we can see That w
ElectronicTransmissionRangeSel == 2 || 4
ElectricParkBrakeApplied == 2
It goes from NEUT --> DRIVE/REV
        Unless VehicleSystemsEnabled ~= 1..
```

VehicleSystemsEnabled does not have a yellow box indicating that its input can be modified in the Flowchart Diagram, if it allowed or even makes sense to basically turn the engine off to achieve this requirement
*/

POSSIBLE TESTS / INPUT SIGNALS
1.) test etrs = drive all the way && parkbrakeApplied = false all the way
- success state
2.) test etrs = reverse all the way && parkbrakeApplied = false all the way
- succes state

3.) test etrs = drive all the way && parkbrakeApplied = True all the way
- expecting fail state (stuck in drive)
4.) test etrs = rev all the way && parkbrakeApplied = True all the way
- expecting fail state (stuck in rev)
5.) test etrs = drive all the way && parkbrakeApplied = Unknown all the way
- expecting fail state (stuck in drive)

6.) test etrs = drive all the way && parkbrakeApplied = True 0<t<10, parkbrakeApplied = False 10<t<35.
- temporal test, expecting succes state for t>10
7.) test etrs = park 0<t<10, etrs = rev 10<t<35 && parkbrakeApplied all the way.
- temporal test, expecitng succes state for t>10

- Had to add conditions to flow chart to make sure stayed in neutral

We have to make sure the gear doesn't just go to DRIVE instead of sticking around in NEUTRAL

Other input variables values
ElectricParkBrakeAvailable
ElectricParkBrakeAvailable == 1
doesn't matter really in this case, but prolly want that in general ;D
ParktoChargeRequested
ParktoChargeRequested == 1
All the way, what gets us from drive/rev towards neutral
Park_BrakePressed
Park_BrakePressed == 1
VehicleSpeed
VehicleSpeed < 0.5
VehicleSpeed == 0.1
HVShutdownRequested - High Voltage Shutdown, condition to force it to go into neutral or park
HVShutdownRequested == 0
Doesn’t rlly matter in this case
VehicleSystemEnabled - is the propulsion system on, can’t go into reverse or drive w/o this, basically shutting off the engine

- Not sure if you can do this since there is no yellow box indicating inputs on the flowchart, but it will go to reverse or drive immediately unless this is == 0.

2.2

If "ElectronicTransmissionRangeSel" indicates PARK or "ParktoChargeRequested" is True and the "ElectricParkBrakeApplied" is False, the PSC shall indicate NEUTRAL via "TransmissionPosition"

// we are checking that the car does not go into park if/until the park brake is applied

//we need to start car in a state other than park, or it will start there indefinitely(ParktoChargeRequested == 1 prevents it going out of park)

- Set ElectronicTransmissionRangeSel == 3 for a few ms, for init, switch it back to == 1 before test starts

POSSIBLE TESTS / INPUT SIGNALS

1.) test etrs = park all the way && ParktoChargeRequested == True all the way && parkbrakeApplied = false all the way

- expecting pass, stay in neutral

2.) test etrs = park all the way && ParktoChargeRequested == True all the way && parkbrakeApplied = True all the way

- expecting fail, go into park

3.) test etrs = park all the way && ParktoChargeRequested == True all the way && parkbrakeApplied = false 0<t<10, parkbrakeApplied = True 10<t<35.

- temporal test, fail for t>10, shift into park

4.) test etrs = drive for 0<t<10, etrs = park 10<t<35 && ParktoChargeRequested == True all the way && parkbrakeApplied = True all the way.

- temporal test, fail for t>10, shift into park

ElectricParkBrakeAvailable

ElectricParkBrakeAvailable == 1

Park_BrakePressed

Park_BrakePressed == 1

Doesnt rlly matter

VehicleSpeed

VehicleSpeed < 0.5 to go into park

VehicleSpeed == 0.1

HVShutdownRequested - High Voltage Shutdown, condition to force it to go into neutral

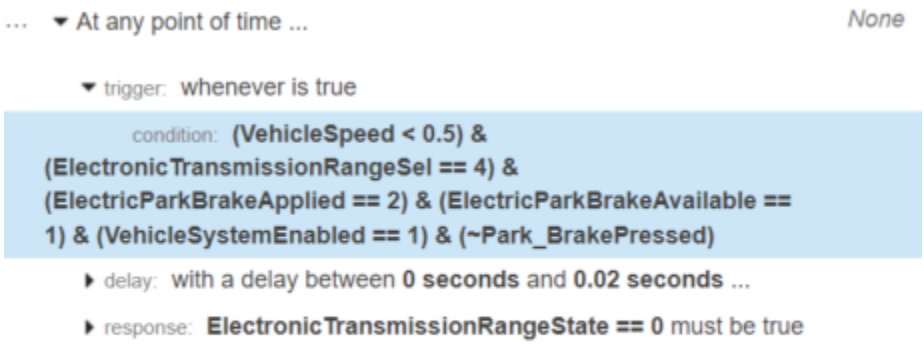
HVShutdownRequested == 1

Prevents it from being in drive or rev

VehicleSystemEnabled - is the propulsion system on, can't go into reverse or drive w/o this, basically shutting off the engine

VehicleSystemEnabled == 1

FAIL: (VehicleSpeed < 0.5) & (ElectronicTransmissionRangeSel == 4) & (ElectricParkBrakeApplied == 2) & (ElectricParkBrakeAvailable == 1) & (VehicleSystemEnabled == 1) & (~Park_BrakePressed)



(VehicleSpeed < 0.5) & (ElectronicTransmissionRangeSel == 4) & (ElectricParkBrakeApplied == 2) & (ElectricParkBrakeAvailable == 1) & (VehicleSystemEnabled == 1) & (Park_BrakePressed)

George Clarifications

- For 2.1, ElectricParkBrakeApplied is supposed to be True, not false
- Only supposed to add logical and temporal assessments, use the given input data, you just need to make the test cases
-