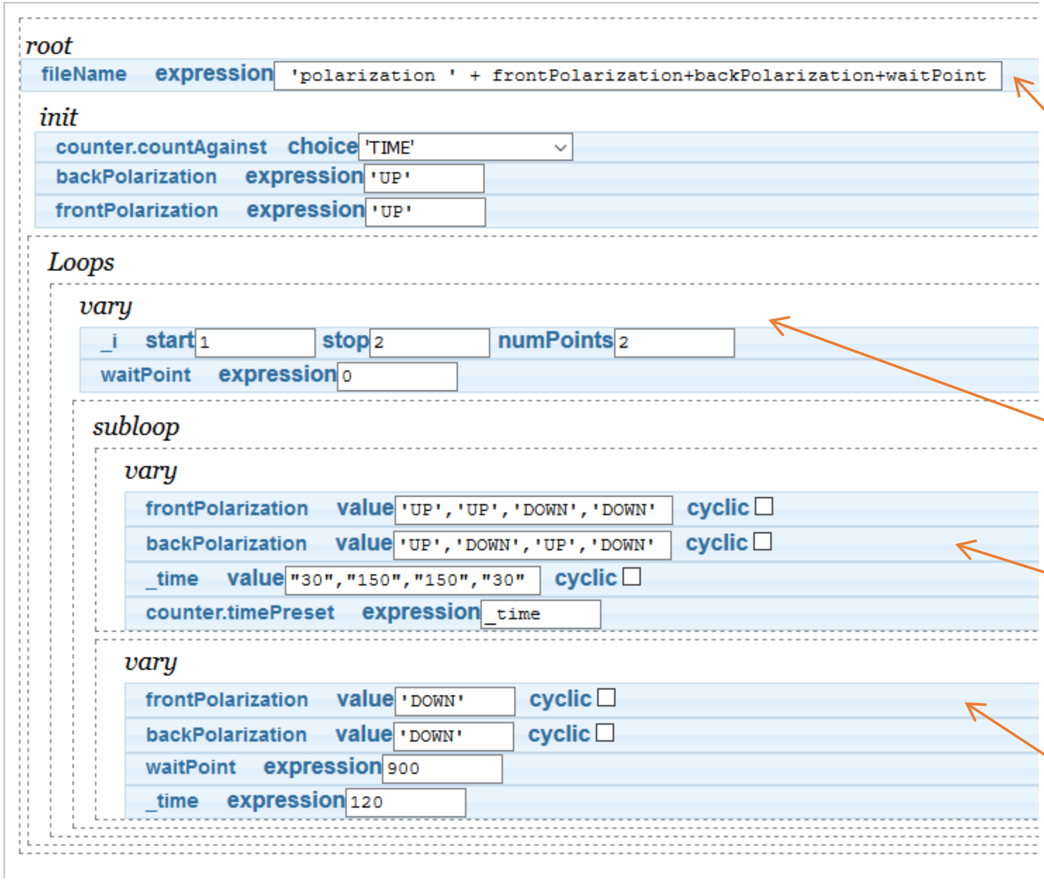

```
root
  fileName  expression  'polarization ' + frontPolarization+backPolarization+waitPoint
  init
    counter.countAgainst  choice  'TIME'
    backPolarization  expression  'UP'
    frontPolarization  expression  'UP'
  Loops
    vary
      _i  start 1  stop 2  numPoints 2
      waitPoint  expression 0
    subloop
      vary
        frontPolarization  value 'UP','UP','DOWN','DOWN'  cyclic ☐
        backPolarization  value 'UP','DOWN','UP','DOWN'  cyclic ☐
        _time  value "30","150","150","30"  cyclic ☐
        counter.timePreset  expression _time
      vary
        frontPolarization  value 'DOWN'  cyclic ☐
        backPolarization  value 'DOWN'  cyclic ☐
        waitPoint  expression 900
        _time  expression 120
```

NOTE: should investigate skipPoint or skipCount for second subloop

Trajectory Comments

Single quotes are text input

Produces 5 files:
polarization_UPUP30
polarization_UPDOWN150
polarization_DOWNUP150
polarization_DOWNDOWN30
polarization_ DOWNDOWN120

spaces produce underscores '_'

Produces 2 points from 1 to 2. At each step, this sets waitPoint to 0 and goes to 1st subloop.

Subloop sets permutation of front pol and back pol. Each permutation counts respective time with no time delay as waitPoint is 0. Goes to next subloop. (see filenames above)

Subloop sets front/back polarization to DOWN state, waits 900s and then counts 120s. Then, goes back to main loop. (This exists because something must move for waitPoint to work.)