

```

def calculateStress() {
  log.debug "[CalculateStress] Running stress evaluation with detailed condition logging..."

  // Ensure devices are not null, and if they are, use default values (0.0 or empty string)
  def dailyHighTemp = dailyHighTempDevice ?
dailyHighTempDevice.currentValue(dailyHighTempAttribute)?.toFloat() : 0.0
  def dailyLowTemp = dailyLowTempDevice ?
dailyLowTempDevice.currentValue(dailyLowTempAttribute)?.toFloat() : 0.0
  def avgHighTemp = avgHighTempDevice ?
avgHighTempDevice.currentValue(avgHighTempAttribute)?.toFloat() : dailyHighTemp
  def avgLowTemp = avgLowTempDevice ?
avgLowTempDevice.currentValue(avgLowTempAttribute)?.toFloat() : dailyLowTemp
  def soilTemp = soilTempDevice ? soilTempDevice.currentValue(soilTempAttribute)?.toFloat() :
0.0
  def uvIndex = uvDevice ? uvDevice.currentValue(uvAttribute)?.toFloat() : 0.0
  def humidity = humidityDevice ? humidityDevice.currentValue(humidityAttribute)?.toFloat() :
0.0

  def tierMap = ["Dormant": 1, "Normal": 2, "Moderate": 3, "High": 4, "Severe": 5]
  def reverseTierMap = tierMap.collectEntries { k, v -> [v, k] }

  def currentTier = "Normal"
  def mowHeight = 1.5
  def bladeHeight = 5
  def irrigationAmount = 0.25

  int dormantMatches = 0, normalMatches = 0, moderateMatches = 0, highMatches = 0,
severeMatches = 0

  // === Dormant Tier
  if (dailyHighTemp < dormantHighTempMax) {
    dormantMatches++; log.info "[Dormant Match] Daily High Temp ${dailyHighTemp} <
${dormantHighTempMax}"
  }
  if (avgHighTemp < dormantAvgHighTempMax) {
    dormantMatches++; log.info "[Dormant Match] 7-Day Avg High Temp ${avgHighTemp} <
${dormantAvgHighTempMax}"
  }
  if (dailyLowTemp < dormantLowTempMax) {
    dormantMatches++; log.info "[Dormant Match] Daily Low Temp ${dailyLowTemp} <
${dormantLowTempMax}"
  }
  if (avgLowTemp < dormantAvgLowTempMax) {

```

```

        dormantMatches++; log.info "[Dormant Match] 7-Day Avg Low Temp ${avgLowTemp} <
${dormantAvgLowTempMax}"
    }
    if (soilTemp < dormantSoilTempMax) {
        dormantMatches++; log.info "[Dormant Match] Soil Temp ${soilTemp} <
${dormantSoilTempMax}"
    }
    if (uvIndex < dormantUVIndexMax) {
        dormantMatches++; log.info "[Dormant Match] UV Index ${uvIndex} <
${dormantUVIndexMax}"
    }

    // === Normal Tier (4 or more matches)
    if (dailyHighTemp < healthyHighTempMax) {
        normalMatches++; log.info "[Normal Match] Daily High Temp ${dailyHighTemp} <
${healthyHighTempMax}"
    }
    if (avgHighTemp < healthyAvgHighTempMax) {
        normalMatches++; log.info "[Normal Match] 7-Day Avg High Temp ${avgHighTemp} <
${healthyAvgHighTempMax}"
    }
    if (dailyLowTemp < healthyLowTempMax) {
        normalMatches++; log.info "[Normal Match] Daily Low Temp ${dailyLowTemp} <
${healthyLowTempMax}"
    }
    if (soilTemp < healthySoilTempMax) {
        normalMatches++; log.info "[Normal Match] Soil Temp ${soilTemp} <
${healthySoilTempMax}"
    }
    if (uvIndex < healthyUVIndexMax) {
        normalMatches++; log.info "[Normal Match] UV Index ${uvIndex} < ${healthyUVIndexMax}"
    }
    if (humidity != null && humidity >= healthyHumidityMin && humidity <= healthyHumidityMax) {
        normalMatches++; log.info "[Normal Match] Humidity ${humidity} in range
${healthyHumidityMin}-${healthyHumidityMax}"
    }

    // === Moderate Tier (any may apply)
    if (dailyHighTemp > moderateHighTempMin) {
        moderateMatches++; log.info "[Moderate Match] Daily High Temp ${dailyHighTemp} >
${moderateHighTempMin}"
    }
    if (avgHighTemp > moderateAvgHighTempMin) {

```

```

        moderateMatches++; log.info "[Moderate Match] 7-Day Avg High Temp ${avgHighTemp} >
${moderateAvgHighTempMin}"
    }
    if (dailyLowTemp > moderateLowTempMin) {
        moderateMatches++; log.info "[Moderate Match] Daily Low Temp ${dailyLowTemp} >
${moderateLowTempMin}"
    }
    if (soilTemp > moderateSoilTempMin) {
        moderateMatches++; log.info "[Moderate Match] Soil Temp ${soilTemp} >
${moderateSoilTempMin}"
    }
    if (uvIndex > moderateUVIndexMin) {
        moderateMatches++; log.info "[Moderate Match] UV Index ${uvIndex} >
${moderateUVIndexMin}"
    }
    if (humidity != null && (humidity < moderateHumidityMin || humidity > moderateHumidityMax))
    {
        moderateMatches++; log.info "[Moderate Match] Humidity ${humidity} outside
${moderateHumidityMin}-${moderateHumidityMax}"
    }

    // === High Tier (2+ matches)
    if (dailyHighTemp > highHighTempMin) {
        highMatches++; log.info "[High Match] Daily High Temp ${dailyHighTemp} >
${highHighTempMin}"
    }
    if (avgHighTemp > highAvgHighTempMin) {
        highMatches++; log.info "[High Match] 7-Day Avg High Temp ${avgHighTemp} >
${highAvgHighTempMin}"
    }
    if (dailyLowTemp > highLowTempMin) {
        highMatches++; log.info "[High Match] Daily Low Temp ${dailyLowTemp} >
${highLowTempMin}"
    }
    if (soilTemp > highSoilTempMin) {
        highMatches++; log.info "[High Match] Soil Temp ${soilTemp} > ${highSoilTempMin}"
    }
    if (uvIndex > highUVIndexMin) {
        highMatches++; log.info "[High Match] UV Index ${uvIndex} > ${highUVIndexMin}"
    }
    if (humidity != null && (humidity < highHumidityMin || humidity > highHumidityMax)) {
        highMatches++; log.info "[High Match] Humidity ${humidity} outside
${highHumidityMin}-${highHumidityMax}"
    }

```

```

// === Severe Tier (3+ matches)
if (dailyHighTemp > severeHighTempMin) {
    severeMatches++; log.info "[Severe Match] Daily High Temp ${dailyHighTemp} >
${severeHighTempMin}"
}
if (avgHighTemp > severeAvgHighTempMin) {
    severeMatches++; log.info "[Severe Match] 7-Day Avg High Temp ${avgHighTemp} >
${severeAvgHighTempMin}"
}
if (dailyLowTemp > severeLowTempMin) {
    severeMatches++; log.info "[Severe Match] Daily Low Temp ${dailyLowTemp} >
${severeLowTempMin}"
}
if (soilTemp > severeSoilTempMin) {
    severeMatches++; log.info "[Severe Match] Soil Temp ${soilTemp} >
${severeSoilTempMin}"
}
if (uvIndex > severeUVIndexMin) {
    severeMatches++; log.info "[Severe Match] UV Index ${uvIndex} > ${severeUVIndexMin}"
}
if (humidity != null && (humidity < severeHumidityMin || humidity > severeHumidityMax)) {
    severeMatches++; log.info "[Severe Match] Humidity ${humidity} outside
${severeHumidityMin}-${severeHumidityMax}"
}

// === Tier Decision Logic
if (severeMatches >= 3) {
    currentTier = "Severe"; mowHeight = 2.5; bladeHeight = 9; irrigationAmount = 0.75
} else if (highMatches >= 2) {
    currentTier = "High"; mowHeight = 2.0; bladeHeight = 7; irrigationAmount = 0.65
} else if (moderateMatches >= 1) {
    currentTier = "Moderate"; mowHeight = 1.75; bladeHeight = 6; irrigationAmount = 0.50
} else if (normalMatches >= 4) {
    currentTier = "Normal"; mowHeight = 1.5; bladeHeight = 5; irrigationAmount = 0.25
    log.debug "[Tier Decision] Normal tier selected based on ${normalMatches} matched
healthy conditions"
} else if (dormantMatches >= 3) {
    currentTier = "Dormant"; mowHeight = 1.25; bladeHeight = 4; irrigationAmount = 0.0
}

log.debug "[CalculateStress] Final Tier Selected: ${currentTier}"

// === Update Child Device ===

```

```

def child = getChildDevice("turfStressMonitorChild")
def smoothedTier = state.smoothedStressLevel ?: currentTier
child?.updateStressData(currentTier, smoothedTier, mowHeight, bladeHeight,
irrigationAmount)

// Save state + timestamped history
state.currentStressTier = currentTier
state.mowHeight = mowHeight
state.bladeHeight = bladeHeight
state.irrigationAmount = irrigationAmount

if (!state.hourlyStressTiers) state.hourlyStressTiers = []

def nowMillis = now()
def numericTier = tierMap[currentTier] ?: 2
state.hourlyStressTiers << [timestamp: nowMillis, tier: numericTier]

def cutoff = nowMillis - (24 * 60 * 60 * 1000)
state.hourlyStressTiers = state.hourlyStressTiers.findAll { it.timestamp >= cutoff }

def localTime = new Date(nowMillis).format("yyyy-MM-dd HH:mm:ss", location.timeZone)
log.debug "[CalculateStress] Logged stress tier ${currentTier} at ${localTime}"
}

```