Data Calculation and Processing Code

Process and calculate data with scripts.

Common Calculations

csharp

```csharp
// Standard Industrial Calculations
public static class IndustrialCalcs
{
    // PID Controller
    public static double PID(double setpoint, double processValue,
                             ref double integral, ref double lastError,
                             double kp, double ki, double kd, double dt)
    {
        double error = setpoint - processValue;
        integral += error * dt;
        double derivative = (error - lastError) / dt;
        lastError = error

        return kp * error + ki * integral + kd * derivative;
    }

    // Moving Average
    public static double MovingAverage(Queue<double> window, double newValue, int windowSize)
    {
        window.Enqueue(newValue);
        if (window.Count > windowSize)
            window.Dequeue();

        return window.Average();
    }

    // Rate of Change
    public static double RateOfChange(double currentValue, double lastValue, double timeInterval)
    {
        return (currentValue - lastValue) / timeInterval;
    }

    // Totalization
    public static double Totalize(double flow, double timeInterval, ref double total)
    {
        total += flow * timeInterval;
        return total;
    }
}
```

Data Processing

python

```python
# Python Data Analysis Script
import numpy as np
import statistics

def analyze_batch_data():
    # Get array data
    temperatures = @Tag.Batch.TemperatureArray
    pressures = @Tag.Batch.PressureArray

    # Statistical analysis
    @Tag.Stats.TempMean = np.mean(temperatures)
    @Tag.Stats.TempStdDev = np.std(temperatures)
    @Tag.Stats.TempMin = np.min(temperatures)
    @Tag.Stats.TempMax = np.max(temperatures)

    # Find outliers
    z_scores = np.abs((temperatures - np.mean(temperatures)) / np.std(temperatures))
    outliers = np.where(z_scores > 3)[0]
    @Tag.Stats.OutlierCount = len(outliers)

    # Correlation
    correlation = np.corrcoef(temperatures, pressures)[0, 1]
    @Tag.Stats.TempPressureCorrelation = correlation

    # Quality determination
    if @Tag.Stats.TempStdDev < 2.0 and len(outliers) == 0:
        @Tag.Batch.Quality = "Good"
    else:
        @Tag.Batch.Quality = "Review Required"
```

In this section...
