NexusEdge Controller

Configuration Examples

HTML Configuration Files & JavaScript Control Logic

Version 1.0.0

Generated: December 27, 2025

AutomataNexus, LLC

Table of Contents

1. Introduction

This document provides example configurations for NexusEdge controller systems. These examples demonstrate the standard format for HTML configuration documentation and JavaScript control logic files used with MegaBas HAT controllers on Raspberry Pi platforms.

Document Types

HTML Configuration Files: Printable documentation sheets containing system overview, I/O assignments, equipment identification, and sequence of operations. These are designed to be printed and kept at the job site for reference.

JavaScript Logic Files: Control logic that runs on the Raspberry Pi controller, implementing PID control, equipment sequencing, safety interlocks, and BMS communication.

2. HTML Configuration Files

HTML configuration files serve as comprehensive documentation for each control system. They are designed to print on standard letter-size paper and include all information needed for installation, commissioning, and maintenance.

2.1 Boiler & Heat Pump Loop System Example

This example shows a dual boiler plant with hot water circulation pumps and a heat pump loop with three circulation pumps. The system uses two MegaBas HAT boards.

Equipment Identification

Equipment	Board Assignment
Boiler-1	MegaBas 0
Boiler-2	MegaBas 0
HW Pump 1	MegaBas 0
HW Pump 2	MegaBas 0
HP Pump 1	MegaBas 1
HP Pump 2	MegaBas 1
HP Pump 3	MegaBas 1

MegaBas 0 I/O Assignments (Boiler Plant)

Channel	Function
TR1	Boiler-1 Enable (24VAC Triac)
TR2	Boiler-2 Enable (24VAC Triac)
TR3	HW Pump 1 Enable (24VAC Triac)
TR4	HW Pump 2 Enable (24VAC Triac)
AO1	Injection Valve (0-10V)
AO2-AO4	Spare
CH1	HW Pump 1 Current (0-10V = 0-50A)
CH2	HW Pump 2 Current (0-10V = 0-50A)
CH3	Boiler Supply Temperature (10K NTC)
CH4	Boiler Return Temperature (10K NTC)
CH5	Outdoor Temperature (10K NTC)
CH6-CH8	Spare

MegaBas 1 I/O Assignments (HP Loop)

Channel	Function
TR1	HP Pump 1 Enable (24VAC Triac)
TR2	HP Pump 2 Enable (24VAC Triac)
TR3	HP Pump 3 Enable (24VAC Triac)
TR4	Bypass Valve Enable (24VAC Triac)
AO1-AO4	Spare
CH1	HP Pump 1 Current (0-10V = 0-50A)
CH2	HP Pump 2 Current (0-10V = 0-50A)
CH3	HP Pump 3 Current (0-10V = 0-50A)
CH4	HP Loop Supply Temperature (10K NTC)
CH5	HP Loop Return Temperature (10K NTC)
CH6-CH8	Spare

Control Features

- Dual boiler lead/lag control with weekly rotation
- Outdoor air temperature reset for boiler supply setpoint
- Injection valve PID control for supply temperature modulation
- 3-pump lead/lag staging for HP loop circulation
- Current monitoring on all pumps for proof of operation
- Automatic failover on pump failure
- Freeze protection mode

2.2 AHU with Steam Bundle System Example

This example shows an air handling unit with chiller and a steam bundle heat exchanger. MegaBas 0 controls the AHU/chiller/CW pumps, while MegaBas 1 controls the steam bundle.

MegaBas 0 I/O Assignments (AHU/Chiller)

Channel	Function
TR1	Fan Enable (24VAC Triac)
TR2	Chiller Enable (24VAC Triac)
TR3	CW Pump 1 Enable (24VAC Triac)
TR4	CW Pump 2 Enable (24VAC Triac)
AO1	OA Damper (0-10V, direct acting)
AO2	Mixed Damper (0-10V, direct acting)
AO3	CW Valve (0-10V, direct acting)
AO4	HW Valve (0-10V, direct acting)
CH1	Supply Temperature (10K NTC)
CH2	Return Temperature (10K NTC)
CH3	Outdoor Temperature (10K NTC)
CH4	Mixed Air Temperature (10K NTC)
CH5	Fan Amps (0-10V = 0-50A)
CH6	Chiller Amps (0-10V = 0-50A)
CH7	CW Pump 1 Amps (0-10V = 0-50A)
CH8	CW Pump 2 Amps (0-10V = 0-50A)

MegaBas 1 I/O Assignments (Steam Bundle)

Channel	Function
TR1	HW Pump 1 Enable (24VAC Triac)
TR2	HW Pump 2 Enable (24VAC Triac)
TR3	CW Booster Pump 1 Enable (24VAC Triac)
TR4	CW Booster Pump 2 Enable (24VAC Triac)
AO1	Steam Valve 1/3 (0-10V, direct acting)
AO2	Steam Valve 2/3 (0-10V, direct acting)
AO3	HW Pump 1 Speed (0-10V VFD)
AO4	HW Pump 2 Speed (0-10V VFD)
CH1	HX Supply Temperature (10K NTC)
CH2	HX Return Temperature (10K NTC)
CH3	HW Pump 1 Amps (0-10V = 0-50A)
CH4	HW Pump 2 Amps (0-10V = 0-50A)
CH5	CW Booster 1 Amps (0-10V = 0-50A)
CH6	CW Booster 2 Amps (0-10V = 0-50A)

3. JavaScript Control Logic Files

Control logic files are JavaScript modules that run on the Raspberry Pi controller. They implement the control algorithms, safety interlocks, and equipment sequencing defined in the sequence of operations.

3.1 AHU with Chiller Control Logic

File Structure

```
// EQUIPMENT IDENTIFICATION
const EQUIPMENT_IDS = {
    AHU_1: 'equipment-unique-id'
};

// CONFIGURATION
const AHU_CONFIG = {
    // Supply Air Temperature Control
    DEFAULT_SUPPLY_SETPOINT: 55,
    MIN_SUPPLY_SETPOINT: 50,
    MAX_SUPPLY_SETPOINT: 65,
    SUPPLY_DEADBAND: 1.0,

    // Outdoor Air Reset
    ENABLE_OA_RESET: true,
    OA_RESET_LOW_TEMP: 40,
    OA_RESET_HIGH_TEMP: 70,

    // Valve PID Control
    CW_VALVE_KP: 8.0,
    CW_VALVE_KI: 0.15,
    CW_VALVE_KD: 0.05,

    // Equipment Thresholds
```

```
    CURRENT_THRESHOLD: 0.5,  // Amps to prove running
};
```

Main Control Function

```
function processAHUControl(data, stateData) {
    // Parse sensor inputs from MegaBas channels
    const supplyTemp = parseSafeNumber(data.CH1, 65);
    const returnTemp = parseSafeNumber(data.CH2, 72);
    const outdoorTemp = parseSafeNumber(data.CH3, 50);
    const mixedAirTemp = parseSafeNumber(data.CH4, 60);

    // Freeze protection check
    if (mixedAirTemp < FREEZE_THRESHOLD) {
        return freezeProtectionMode();
    }

    // Calculate setpoint with OA reset
    const supplySetpoint = calculateOAResetSetpoint(outdoorTemp);
    const supplyError = supplyTemp - supplySetpoint;

    // PID control for valves
    if (supplyError > DEADBAND) {
        // Cooling mode - modulate CW valve
        cwValvePosition = calculatePID(supplyError, cwPidState);
    } else if (supplyError < -DEADBAND) {
        // Heating mode - modulate HW valve
        hwValvePosition = calculatePID(Math.abs(supplyError), hwPidState);
    }

    // Return outputs for MegaBas board
    return {
        TR1: fanEnable,
        TR2: chillerEnable,
        TR3: cwPump1Enable,
        TR4: cwPump2Enable,
        AO1: oaDamperVoltage,
        AO2: mixedDamperVoltage,
        AO3: cwValveVoltage,
        AO4: hwValveVoltage,
        controlMode: controlMode,
        lastUpdate: new Date().toISOString()
    };
}
```

3.2 Steam Bundle Heat Exchanger Control

Dual Valve Staging Logic

```
// Dual modulating valve control (1/3 and 2/3 capacity)
if (Math.abs(hxError) > HX_DEADBAND) {
    controlMode = 'heating';

    // Primary valve (1/3) - PID control
    valve1Position = calculatePID(hxDemand, valve1PidState);

    // Secondary valve (2/3) - stages on at high demand
    if (valve1Position > VALVE_2_ENABLE_THRESHOLD) {
        valve2Position = calculatePID(hxDemand - 20, valve2PidState);
    }
}

// Lead/lag pump control with VFD speed
if (valve1Position > 5 || valve2Position > 5) {
    if (leadPump === 1) {
        hwPump1Enable = true;
        if (valve1Position > LAG_PUMP_THRESHOLD) {
            hwPump2Enable = true;  // Stage lag pump
        }
    }
}

// Weekly lead/lag rotation
if (timeSinceRotation >= ROTATION_INTERVAL) {
```

```
    leadPump = leadPump === 1 ? 2 : 1;
}
```

4. I/O Assignment Standards

The following standards are used for MegaBas HAT I/O assignments across all NexusEdge controllers.

Triac Outputs (TR1-TR4)

Type	Voltage	Use Case
Equipment Enable	24VAC	Fans, pumps, chillers, boilers
Valve Control	24VAC	On/off valves, bypass valves
Damper Control	24VAC	On/off dampers

Analog Outputs (AO1-AO4)

Type	Signal	Use Case
Valve Position	0-10V	Modulating valves (CW, HW, steam)
Damper Position	0-10V	Modulating dampers (OA, RA, mixed)
VFD Speed	0-10V	Variable frequency drives

Analog Inputs (CH1-CH8)

Type	Signal	Use Case
Temperature	10K NTC Type 2	Supply, return, outdoor, mixed air
Current	0-10V (0-50A)	Motor amp monitoring for proof
Pressure	0-10V	Duct or pipe pressure (if needed)
Humidity	0-10V	Space or duct humidity (if needed)

5. Control Logic Patterns

PID Control

All modulating outputs use PID control with anti-windup. The PID function calculates proportional, integral, and derivative components based on the error signal.

```
function calculatePID(error, pidState, kp, ki, kd, maxIntegral) {
    const dt = (currentTime - pidState.lastTime) / 1000;

    // Proportional
    const P = kp * error;

    // Integral with anti-windup
    pidState.integral += error * dt;
    pidState.integral = clamp(pidState.integral, -maxIntegral, maxIntegral);
    const I = ki * pidState.integral;

    // Derivative
    const D = kd * ((error - pidState.previousError) / dt);

    return clamp(P + I + D, 0, 100);
}
```

Lead/Lag Equipment Staging

Redundant equipment (pumps, boilers) uses lead/lag staging with automatic weekly rotation to ensure equal runtime and equipment longevity.

```
// Stage lag equipment based on demand
if (valve1Position > LAG_ENABLE_THRESHOLD) {
    lagEquipmentEnable = true;
}

// Weekly rotation (every Monday at 2:00 AM)
const ROTATION_INTERVAL = 7 * 24 * 60 * 60 * 1000;  // 7 days
if (currentTime - lastRotation >= ROTATION_INTERVAL) {
    leadEquipment = leadEquipment === 1 ? 2 : 1;
    lastRotation = currentTime;
}

// Failover if lead fails
if (leadEquipmentEnabled && !leadEquipmentProven) {
    // Switch to lag as new lead
    lagEquipmentEnable = true;
    generateAlarm('LEAD_EQUIPMENT_FAILURE');
}
```

Safety Interlocks

Condition	Action
Mixed air temp < 40F	Freeze protection: close OA damper, open HW valve
Supply temp > 190F	High limit: close steam valves, alarm
Pump enabled, no current > 30s	Pump failure: switch to lag, alarm
Fan not proven	Disable chiller (interlock)
I2C communication loss	Safe state: all outputs off, alarm

For more information, visit nexusconnectbridge.automatanexus.com or contact AutomataNexus, LLC.