

```

description = "Custom CCP_VFD2"
notes = "Custom for CCP and VFD"
vendor = "User"
vendorUrl = ""

//
=====
===
// GLOBAL SETUP PARAMETERS
//
=====
===
var state = {
    currentSpindleSpeed: -1,
    currentTool: -1,
    xval: null,
    yval: null,
    zval: null,
    fval: null,
    sval: null,
    tval: null,
    motionMode: null,
    rpm: 0,
    wasHomeZ: false,
    lastTool: -1,
    spindle: false
};

var CONFIG_DWELL_SECONDS = 10; // Adjust VFD warmup delay time here
var outputMetric = true;
var extension = "nc";

//
=====
===
// HELPER UTILITIES
//
=====
===
function createFormatter(name, numDigits, useUnits) {
    var lastOutput = "";
    return {
        format: function(val) {
            if(val == undefined) {
                this.reset();
                return "";
            }
            if(!outputMetric && useUnits) {
                val = val / 25.4;
            }

```

```

        var fmt = val.toFixed(numDigits);
        var ret = "";
        if(fmt != lastOutput) {
            ret += name;
            ret += fmt;
            lastOutput = fmt;
        }
        return ret;
    },
    reset: function() {
        lastOutput = "--";
    }
};
}

function formatUnits(val) {
    var un = "mm";
    var digits = 2;
    if(!outputMetric) {
        val /= 25.4;
        un = "in";
        digits += 1;
    }
    return "" + val.toFixed(digits) + un;
}

function formatDimension(val) {
    var digits = 2;
    if(!outputMetric) {
        val /= 25.4;
        digits += 1;
    }
    return "" + val.toFixed(digits);
}

function outputSafeZ() {
    if(state.wasHomeZ) {
        return;
    }
    onComment("Safely lifting Z incrementally to avoid workholding");
    // FIX: Using incremental relative moves instead of G53 machine
coordinates to avoid Out of Range errors
    writeln("G91");
    if(outputMetric) {
        writeln("G0 Z12.0"); // Lift up 12mm
    } else {
        writeln("G0 Z0.5"); // Lift up 0.5 inches
    }
    writeln("G90"); // Force back to absolute coordinate space

```

```

    state.motionMode.reset();
    state.zval.reset();
    state.wasHomeZ = true;
}

function writeBlock(){
    var ln = "";
    for (var i = 0; i < arguments.length; i++) {
        ln += arguments[i];
    }
    writeLn(ln);
}

function onComment(text) {
    writeLn("(" + text + ")");
}

//
=====
===
// CORE INTERFACE LIBRARIES
//
=====
===
function onOpen() {
    var coordDigits = outputMetric ? 3 : 4;

    state.xval = createFormatter("X", coordDigits, true);
    state.yval = createFormatter("Y", coordDigits, true);
    state.zval = createFormatter("Z", coordDigits, true);
    state.fval = createFormatter("F", 1, true);
    state.sval = createFormatter("S", 0);
    state.tval = createFormatter("T", 0);
    state.motionMode = createFormatter("G", 0);
    state.rpm = 0;
    state.wasHomeZ = false;
    state.lastTool = -1;
    state.spindle = false;

    onComment("Design File: " + sourceFilename);
    onComment("stockMin:" + formatUnits(stockMinX) + ", " +
formatUnits(stockMinY) + ", " + formatUnits(stockMinZ));
    onComment("stockMax:" + formatUnits(stockMaxX) + ", " +
formatUnits(stockMaxY) + ", " + formatUnits(stockMaxZ));

    var length = stockMaxX - stockMinX;
    var width = stockMaxY - stockMinY;
    var height = stockMaxZ - stockMinZ;
    onComment("STOCK/BLOCK," + formatDimension(length) + ", " +
formatDimension(width) + ", " + formatDimension(height) + ", " +

```

```

formatDimension(-stockMinX) + ", " + formatDimension(-stockMinY) + ",
" + formatDimension(-stockMinZ));

    writeln("G90");
    if(outputMetric) {
        writeln("G21");
    } else {
        writeln("G20");
    }

    outputSafeZ();
    writeln("*** SYSTEM START - OPTIMIZED FOR UGS ***");
    writeln("(Check initial machine alignment state)");
}

function onSection() {
    var tnum = toolNumber;
    var force = forceToolChange;

    // 1. MANUAL TOOL CHANGE SEQUENCE INTERCEPT
    if (tnum !== state.currentTool) {
        state.currentTool = tnum;
        state.lastTool = tnum;

        writeln("M05"); // Turn off VFD spindle before tool change

        // FIX: Replaced G53 Z-5.0 tool clearance with safe
incremental relative clearing lift
        writeln("G91");
        if(outputMetric) {
            writeln("G0 Z25.0"); // Lift up 25mm for clearance
        } else {
            writeln("G0 Z1.0"); // Lift up 1 inch for clearance
        }
        writeln("G90");

        writeln("(MSG, ACTION REQUIRED: Load Tool #" + tnum + ")");
        writeln("(UGS: Fetching coordinates to console)");
        writeln("M00"); // Pause machine execution for physical change
        writeln("*** VFD SPINUP SEQUENCE ***");
        writeBlock("M03", state.sval.format(s));
        writeln("G04 P" + CONFIG_DWELL_SECONDS + " (Dwell for VFD
ramp stability)");
    }

    // 2. ORIGINAL GEOMETRY TOOLPATH HANDLING
    if (typeof(Geometry) !== "undefined") {
        if(tnum !== state.lastTool || force == true) {
            state.lastTool = tnum;

```

```

        onSpindleSpeed(0);
        outputSafeZ();

        // Cutviewer Tool String
        onComment("TOOL/MILL," + formatDimension(toolDiameter) +
            ", " + formatDimension(toolCornerRadius) + ", " +
            formatDimension(toolFluteLength) + ", " +
            Number(toolAngle).toFixed(2));

        state.tval.reset();
        writeBlock("M6", state.tval.format(tnum));

        state.xval.reset();
        state.yval.reset();
        state.zval.reset();
        state.fval.reset();
        state.wasHomeZ = true;
    }
}

function onSpindleSpeed(s) {
    if(s <= 0) {
        state.spindle = false;
        state.currentSpindleSpeed = 0;
        writeBlock("M05");
    } else {
        state.spindle = true;

        if (s !== state.currentSpindleSpeed) {
            state.currentSpindleSpeed = s;
            state.sval.reset();

            writeLn("*** VFD SPINUP SEQUENCE ***");
            writeBlock("M03", state.sval.format(s));
            writeLn("G04 P" + CONFIG_DWELL_SECONDS + " (Dwell for VFD
ramp stability)");
        }
        state.rpm = s;
    }
}

function onClose() {
    writeLn("*** JOB COMPLETION SAFETY CODES ***");
    writeLn("M05 (Turn off VFD Spindle immediately)");

    // FIX: Removed dangerous absolute G53 retract block. Swapped for
    deterministic incremental end lift.
    writeLn("G91");
    if(outputMetric) {

```

```

        writeLn("G0 Z20.0 (Clear tool up away from stock surface)");
    } else {
        writeLn("G0 Z0.75 (Clear tool up away from stock surface)");
    }
    writeLn("G90");

    if (outputMetric) {
        writeLn("G21");
    } else {
        writeLn("G20");
    }

    writeLn("M30");
}

function onLinear(x, y, z, feed) {
    if(state.wasHomeZ) {
        onComment("PREPOSITION FOR PLUNGE");
        writeBlock(state.motionMode.format(1), state.xval.format(x),
state.yval.format(y), state.fval.format(feed));
        state.wasHomeZ = false;
    }
    var xv = state.xval.format(x);
    var yv = state.yval.format(y);
    var zv = state.zval.format(z);

    if(xv != "" || yv != "" || zv != "") {
        writeBlock(state.motionMode.format(1), xv, yv, zv,
state.fval.format(feed));
    }
}

function onRapid(x, y, z) {
    if(state.wasHomeZ) {
        onComment("PREPOSITION FOR RAPID PLUNGE");
        writeBlock(state.motionMode.format(0), state.xval.format(x),
state.yval.format(y));
        state.wasHomeZ = false;
    }
    writeBlock(state.motionMode.format(0), state.xval.format(x),
state.yval.format(y), state.zval.format(z));
}

function onTerminate() {
    // Teardown step
}

```