plcLib 2.0 Beta 8 Release Notes

Added 3 new target systems.

- Automation 2040 W (Pimoroni).
- Automation 2040 W Mini (Pimoroni).
- Inventor 2040 W (Pimoroni).

Added new *Motors* example menu.

- Added direct support for hobby servos via a new *Servo* class and associated *sout* command. See examples: *Motors* > *ServoPotentiometer* and *Motors* > *ServoSwitch*. (Arduino sketches make use of the built-in *Servo* library, with some syntax differences.)
- Added support for DC motors with directional and PWM-based speed control via a new *Hbridge* class. See examples: *Motors* > *HbridgePWM*, *Motors* > *DualHbridgePWM*, *Motors* > *HbridgeOnOff*.
- Added support for pulse-based position encoders via the *PositionEncoder* class. See example: *Motors* > *PositionEncoder*.
- Added support for stepper motors via a new *Sequencer* class and associated 1-dimensional data arrays. See examples: *Motors* > *StepperHalfStep*, *Motors* > *StepperFullStep*, *Motors* > *StepperWaveStep*, *Motors* > *StepperContinuous*, *Motors* > *StepperPosition*.
- Removed previous hobby servo examples from *Extras* folder, as servos are now directly supported.

Updated *Variables* folder and examples.

- Renamed/reworded existing examples based on auxiliaries. See *Variables* > *LatchCommandAuxiliary*, *Variables* > *SetResetAuxiliary*, *Variables* > *SetResetAuxiliaryBuilt-in*.
- Added new example with fixed logic auxiliaries, named *logic1* and *logic0*. See example: *Variables* > *FixedLogic*.
- Added the ability to define *global variables* in the JavaScript editor, using the syntax '*variable_name = positive_integer_value;*'. This is in addition to the ability to define a 1-dimensional array, as used by the Sequencer class. See example: *Variables* > *ConditionalCode*.
- Added the ability use variables to customise internal plcLib parameters (this is an ongoing development area). See example: *Variables* > *CustomBaudRate*.

Modified existing commands

- Modified the *pout* (PWM output) command to use either an automatically scaled value, or a fixed PWM value (0-255). This mechanism is also offered by the newly introduced *sout* (Servo out) command, with allowed output angles in the range 0-180°.

Modified Arduino pin mapping

- Changed output pin mappings for the *Arduino Uno (default)* option, to give PWM capability on all digital outputs *Y0-Y3* (pins 2, 3, 4, 5 changed to 3, 5, 6, 9).

Back-end enhancements

- The target system code text area is now automatically cleared when the user loads a new example or changes the target system.
- The code generation process (JavaScript to C++) has been enhanced, to support more complex scenarios (including 3rd party library direct support). As part of the associated changes, the *header* folder associated with the live simulation has been replaced by the *target* folder.
- The display of certain visual elements is suppressed when the plcLib editor is displayed within an iframe (pull-down menu, code generation area). This experimental feature is intended to support the future inclusion of live examples within web pages and online documentation.
- The majority of internal JavaScript debugging (*console.log*) commands have been commented out, pending permanent removal in a future version.
- The unused *Pin* class has been removed from the Arduino library version.

Bug fixes

- Corrected an error where some Boolean logic commands did not work correctly when logic inputs were *states*. This fixes an error in the example: *Apps* > *SingleTrafficLight* when running on an Arduino.

Last updated 21 September 2025 by WD.