# Overview

This document details the use of software tool MPLAB® Integrated Programming Environment (IPE) for programming CAN Servo-Nodes. The IPE provides a method to perform programming and serial number control of embedded software. The IPE is a free tool, which is downloaded along with the MPLAB IDE or can be downloaded independently (<http://microchip.wikidot.com/ipe:installation>).

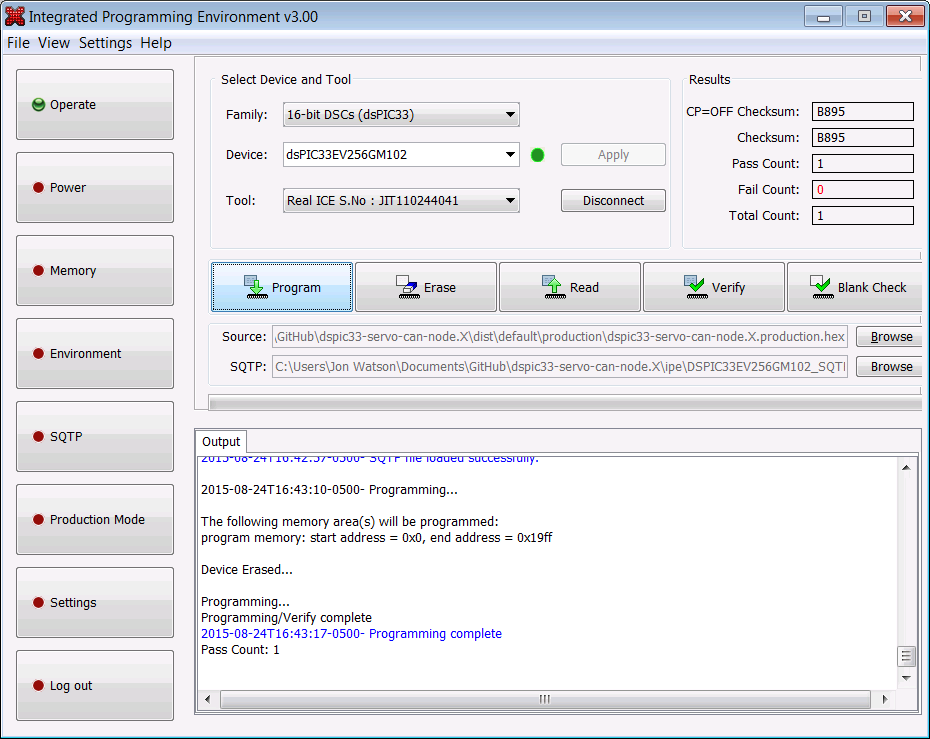
Steps listed below as “***optional***” are needed if a unique serial number for each CAN Servo-Node is desired. The IPE maintains a Serialized Quick Turn Programming (SQTP) file for recording used and available serial numbers. This SQTP file is updated by the IPE tool during the programming operation to identify that a serial number was used to program a target. The SQTP file must be maintained within a Version Control System (VCS) to guarantee that unique serial numbers are maintained for different target devices.

Different sections exist below for ‘initial programming’ and ‘reprogramming’. The difference exists because the serial number is programmed during initial programming, and then preserved within non-volatile memory when reprogramming the device. If unique serial numbers are not required the ‘initial programming’ steps can always be performed, even when reprogramming a device.

# Initial Programming Steps

1. Open the MPLAB IPE tool.
2. Select: *Settings >> Advanced Mode*.
3. Enter password (‘microchip’ is default) and press: *Log on*.
4. On the left side of the Screen, select: *Operate*.
5. From the “Family” drop-down, select: *16-bit DSCs (dsPIC33).*
6. From the “Device” drop-down, select: *dsPIC33EV256GM102*.
7. From the “Tool” drop-down, select the programming tool being used to program the target.
8. Click the button: *Apply*.
9. Click the button: *Connect*.
10. If presented with the cautionary pop-up window on device voltage, select: *Yes*.
11. To the right of the “Source” label select button: *Browse*.
12. Navigate to and select the required file to download to the target, for example:  
    *C:\Users\Jon Watson\Documents\GitHub\dspic33-servo-can-node.X\dist\default\production\dspic33-servo-can-node.X.production.hex*
13. (***optional***) To the right of the “SQTP” label select button: *Browse*.
14. (***optional***) Select the SQTP file which includes serial number information, for example:  
    *C:\Users\Jon Watson\Documents\GitHub\dspic33-servo-can-node.X\ipe\DSPIC33EV256GM102\_SQTP.num*
15. Program the target by selecting button: *Program*.
16. (***optional***) Update the SQTP file within a selected VCS.

The initial programming operation is complete. The below figure provides an example screenshot of the MPLAB IPE tool following the programming steps being completed:



# Reprogramming Steps (***optional***)

1. Open the MPLAB IPE tool.
2. Select: *Settings >> Advanced Mode*.
3. Enter password (‘microchip’ is default) and press: *Log on*.
4. On the left side of the Screen, select: *Memory*.
5. In the “Preserve Memory” section, check the check-box: *Preserve Flash on Program*.
6. Select for a “Start Address”: *0x200*.
7. Select for an “End Address”: *0x2FF*.
8. On the left side of the Screen, select: *Operate*.
9. From the “Family” drop-down, select: *16-bit DSCs (dsPIC33).*
10. From the “Device” drop-down, select: *dsPIC33EV256GM102*.
11. From the “Tool” drop-down, select the programming tool being used to program the target.
12. Click the button: *Apply*.
13. Click the button: *Connect*.
14. If presented with the cautionary pop-up window on device voltage, select: *Yes*.
15. To the right of the “Source” label select button: *Browse*.
16. Navigate to and select the required file to download to the target, for example:  
    *C:\Users\Jon Watson\Documents\GitHub\dspic33-servo-can-node.X\dist\default\production\dspic33-servo-can-node.X.production.hex*
17. Program the target by selecting button: *Program*.

The reprogramming operation is complete. The below figure provides an example screenshot of the MPLAB IPE tool following the programming steps being completed (Note: from the “Output” window it can be verified that the expected memory range was preserved):

