

FROM: M. McClung, 76336,3110  
TO: Dave Brown, 72103,2235  
DATE: 9/19/94 10:39 PM

Re: WOSA/XMC, Draft 2, Section 3.3

Hello Dave,

For this correspondence I would like to brainstorm Section 3.3 that describes a scenario for code generation as it relates to "Initialization Tuning". However, I am not familiar with with the concept of "Initialization Tuning" in our products. Also, I wasn't sure which MCAPI code generation interface this refers to, although I did see an Initialize method in the ICurrentState interface.

While we're on the subject of code generation interfaces, I think there is a better way of implementing code generation than using code generation interfaces. How about giving each interface the capability of doing code generation. For example, the C code from Section 4 (pMove->Interpolated(3, ad)) could either perform the interpolated move or generate code for the interpolated move and save it to a file. At the MCAPI level code generation could be turned on/off via a flag passed to an interface method.

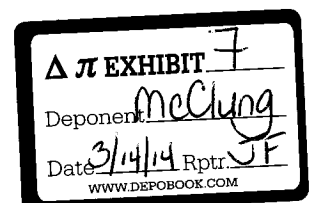
Also, in our products there would be 2 kinds of code generation. First, there is program code generation. For example,

```
DEF PROG1 ;Define a program
D100,200 ;
GO11 ;Move to XY=100,200
OUT11101 ;Turn some outputs on/off
T2 ;Wait 2 seconds
END ;End program definition
```

```
RUN PROG1 ;Execute program
```

The second kind of code generation would not generate the DEF, END and RUN statements. The code would be a stream of commands meant to be executed

RGB00051658



ROY-G-BIV CORPORATION  
EXHIBIT 2021-7  
ABB v ROY-G-BIV  
TRIAL IPR2013-00062

on the fly one of our control products. I call this on-the-fly code generation.

Rather than send generated code to a Windows Registry, I would like to allow the user or programmer to specify where the resultant code should go. The user or programmer might also want to generate code to several files in one application.

With regards to initialization, will WOSA/XMC provide guidelines as to what should be included in an Initialize function at the MCSPi level? Your example in Section 3 suggests some sort of axis-specific initialization via an axis data structure. I'm assuming this structure holds info like axis scaling, encoder resolution, drive resolution (steppers), end-of-travel and home limit settings, default motion, etc. Our products also have system wide initialization such as I/O definitions, memory partitions, performance parameters, etc. Furthermore, there is special initialization for special features such as contouring, following, registration, joystick, etc.

RGB00051659