W ‘20

Processing Sheet Macro Report

MFGE 495: DIRECTED RESEARCH PROJECT

JACK WILSON

Contents

Abstract

Process sheets are important for documenting the sequence of operations used to machine a component. They include information on the cutting tools, process parameters, and a sketch to properly configure a CNC machine before executing a program. Currently, these sheets are created manually by students taking CNC courses and by those completing projects using the MFGE program's CNC laboratory. The goal of this project is to write a macro script to automatically generate these documents. Using this method of creation, the efficacy of producing these documents and the consistency of them can be improved. Using VBA, the PPR tree of a prismatic machining document will be scanned for information which will then be extracted into a formatted Excel document.

Progress Report

The process of generating the macro was tedious and meticulous at first, but after some major breakthroughs, it was created by identifying and solving coding problems. Over this period, my knowledge of the CATIA structure and understanding of Visual Basics were improved substantially. The first couple weeks were spent working through a similar provided example. Through this example, I was able to identify how to use and interact will Excel as well as some of the basic CATIA commands. It was identified that there wasn't much interaction with the machining documents and was difficult to identify many solutions towards extracting any of the machining information. Most information gathered by this method were in specified parameters under the PPR tree or were manually input by the user with user forms.

The next approach taken was examining a built-in CATIA script that generates an HTML documentation of the whole machining process, including information about the tools, images of the first setup, and machining parameters of each manufacturing activity. After finding the location of this script, I ported it from a .catvbs to a .catvba module. From there, I was able to read it as a library using the built-in VBA editor in CATIA. On top of the 4000 lines of code in this document, there was the added challenge that most of the documentation and comments in the file were written in French. Using previous studies of the language with assistance from a translator, I was able better comprehend the code compared to reading line by line through each function. After spending a couple of weeks understanding the script, I extracted the necessary methods needed to approach solving multiple problems, including the methodology for taking screenshots, obtaining machining information, parsing, and converting methods. From there, I commented out unnecessary functions and removed irrelevant code.

By combining the excel process used from the example script and the methods acquired from the CATIA document generation, a working example started to form. At this point, more problem solving was required by investigation the structure of CATIA to manipulate niche tools that aren't typically used by modelers, many of which come down to a checkbox through a series of menus. For example, figuring out a way to "create a sketch" for each part setup proved to be a challenge (see Figure 1). To be as close as possible to the manually made operation documents, the view of the part had to be captured similar to isometric views in the drafting workbench of CATIA, with specific zooms, orientations, and without defects (e.g. highlighted objects, visible planes and sketches). To solve this specific problem, I figured out how to change the CATIA view and settings manually, found out the corresponding locations and values in the active view and settings using debugging tools, then programmatically implement these features alongside consultation from the documentation. Additionally, user interaction was needed to properly orient the model and select specific machine setups. A user form was used to prompt the user to select a specified machine set up and to orient the model.

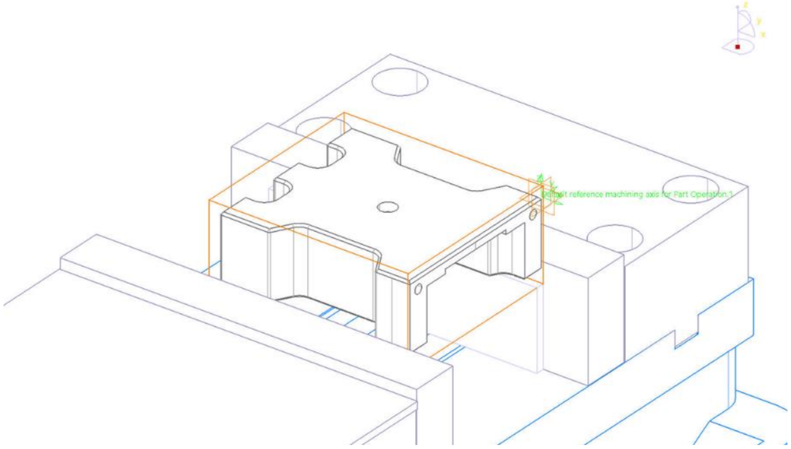


Figure 1 - Output photo of the setup

Furthermore, there were several challenges to formatting the Excel sheet to display information in a professional manner. Major problems that were tackled in this section include formatting as a table, fitting to one page lengthwise, and gathering user inputs. The difficulty of working with Excel, is that it contains an entirely separate library of tools. Every Excel function had to be called out leading with *Excel.<parameter/function>* or with a *With Excel* statement. Moreover, Excel has its own set of enumerators that represented integer values (e.g. *xlYes=1*, *xlHorizontal=-4128*) of which could not be called out by name. Therefore, extra time was spent attempting to discover these values from their corresponding libraries. Additionally, the use of a template was avoided to create a minimal dependency script (i.e. there are no references to other files like templates). Extra time taken for precision was dedicated to the proper formatting of the document, approximating distances programmatically to improve readability while fitting on a single page in the lengthwise direction. Lastly, more user information was gathered for documentation purposes, including name, class, lab time, and product name, using a modal user-form. Combined, a processing sheet was formed and approved. Some recommended changes for the future have been proposed before implementing into a MFGE course at WWU. These are described in the Future Improvements section.

Features

As of its current state, the macro produced has a variety of set features. More options and customizations may become available in the future, as this project was designed to be edited to add extra parameters and features. Descriptions of the current features are described:

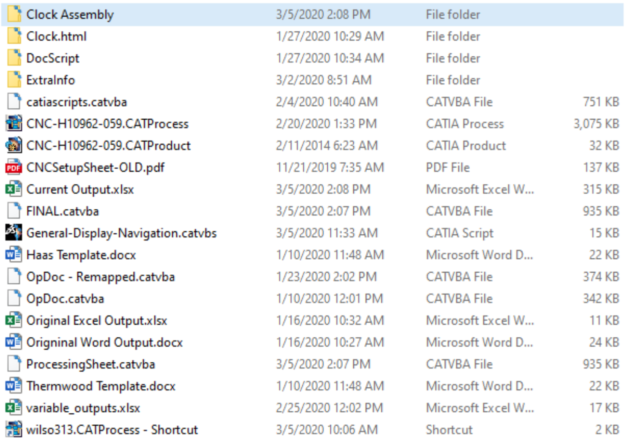
- Easy to run – the macro is designed to be run from a macro library but can be implemented to run via a button.
- User interaction – the user is required to input certain information pertaining to the CATIA document as well as provide views for screenshots.
- Background processes – all the documentation is generated in the background apart from user interactions
- Machining information – a large output of machining information is produced:
    - Activity – a short description of the manufacturing activity
    - Tool # – the tool number associated with the manufacturing activity
    - Tool Description – a short description of each the type of tool is provided
    - Feed – the general feed rate of the of the manufacturing activity
    - Spindle Speed – the spindle speed of the operation
    - Approach Feed – the feed rate of the approach of the operation
    - Retract Feed – the feed rate of the retract of the operation
    - Finishing Feed – the feed rate associated with a finishing operation
    - Machining Time – the time spent cutting into material
    - Total Time – the time spent in an operation
- Sketch representation – a sketch of the part setup is provided for reference and dimensioning

Manual

A short manual is provided for implementation and use of the macro. A description of the file structure is provided as well as quick step by step instruction of using the macro script.

File Structure

A brief description of the files associated with the folder for the project have been described for navigational purposes:



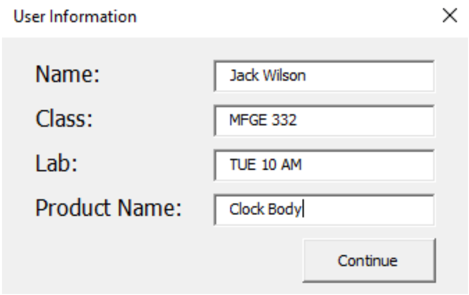
Clock Assembly	3/5/2020 2:08 PM	File folder	
Clock.html	1/27/2020 10:29 AM	File folder	
DocScript	1/27/2020 10:34 AM	File folder	
ExtraInfo	3/2/2020 8:51 AM	File folder	
catiascripts.catvba	2/4/2020 10:40 AM	CATVBA File	751 KB
CNC-H10962-059.CATProcess	2/20/2020 1:33 PM	CATIA Process	3,075 KB
CNC-H10962-059.CATProduct	2/11/2014 6:23 AM	CATIA Product	32 KB
CNCSetupSheet-OLD.pdf	11/21/2019 7:35 AM	PDF File	137 KB
Current Output.xlsx	3/5/2020 2:08 PM	Microsoft Excel W...	315 KB
FINAL.catvba	3/5/2020 2:07 PM	CATVBA File	935 KB
General-Display-Navigation.catvbs	3/5/2020 11:33 AM	CATIA Script	15 KB
Haas Template.docx	1/10/2020 11:48 AM	Microsoft Word D...	22 KB
OpDoc - Remapped.catvba	1/23/2020 2:02 PM	CATVBA File	374 KB
OpDoc.catvba	1/10/2020 12:01 PM	CATVBA File	342 KB
Original Excel Output.xlsx	1/16/2020 10:32 AM	Microsoft Excel W...	11 KB
Origninal Word Output.docx	1/16/2020 10:27 AM	Microsoft Word D...	24 KB
ProcessingSheet.catvba	3/5/2020 2:07 PM	CATVBA File	935 KB
Thermwood Template.docx	1/10/2020 11:48 AM	Microsoft Word D...	22 KB
variable_outputs.xlsx	2/25/2020 12:02 PM	Microsoft Excel W...	17 KB
wilso313.CATProcess - Shortcut	3/5/2020 10:06 AM	Shortcut	2 KB

Figure 2 - File system associated with the project

- Clock Assembly – contains the clock project from MFGE 332 and is used for testing purposes
- Clock.html – the machining documentation autogenerated from CATIA’s documentation command
- DocScript – contains a copy of CATIA’s documentation command and the associated files
- ExtraInfo – stores screenshots and relevant resources for some problem-solving purposes
- catscripts.catvba – contains the converted library from CATIA’s documentation macro
- CNC-H10962-059.CATProcess – a process that was provided for testing purposes
- CNC-H10962-059.CATProduct – the associated product file for the process above
- CNCSetupSheet-OLD.pdf – the output of the example provided
- Current Output.xlsx – the current output of the created macro
- FINAL.catvba – the macro library of the final state of the program
- General-Display-Navigation.catvbs – a copy of the macro script CATIA uses for its documentation command
- Haas Template.docx – the template file for the example macro
- OpDoc – Remapped.catvba – contains a remapped version of the OpDoc macro library used for testing at home
- OpDoc.catvba – contains the macro library of the example provided
- Original Excel Output.xlsx – contains the excel output of the macro provided
- Original Word Output.docx – contains the finished output of the macro provided
- ProcessingSheet.catvba – contains the macros for the project
- Thermwood Template – template for the sample macro provided
- variable_outputs.xlsx – contains all the variable that are spit out by CATIA’s macro
- wilso313.CATProcess – Shortcut – a shortcut to reach the CATProcess file faster

Step-by-Step Instructions

1) Open a prismatic machining (.CATProcess) file in CATIA V5.
2) Navigate to the macro.
    a) Tools > Macros > Macros… (Alt + F8)
    b) If the library doesn’t exist add it. A copy of this file is found at N:\Research\AMRL\CAM Automation\Process Sheet Automation\FINAL.catvba
    c) Select ProcessingSheet and run. Make sure any stock is visible before running, as it will be inaccessible during the operation.
3) Run the macro.
    a) DO NOT click into excel at any point during this process, if it appears.
    b) A user form will appear. Fill out the name, class, lab, and product name sections appropriately.

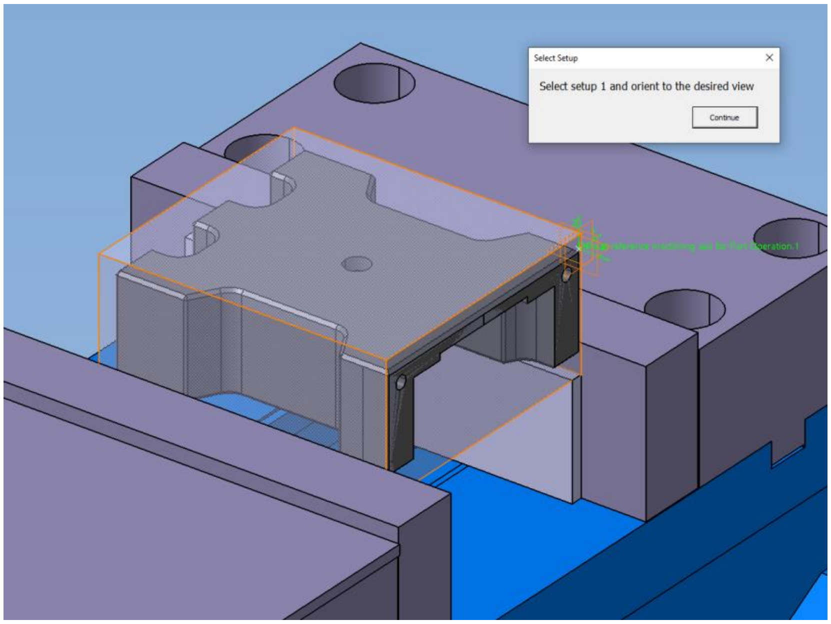


Name: Jack Wilson
Class: MFGE 332
Lab: TUE 10 AM
Product Name: Clock Body

Figure 3 - User-form for information

c) A notification will appear and prompt the user to select a setup and orient the model. Select the mentioned machine setup, orient the model, then select the stock associated with it so it is highlighted.

i) NOTE: the stock is the only thing that should be highlighted.



Select setup 1 and orient to the desired view

Figure 4 - Example orientation with stock highlighted

d) Repeat part c until an Excel file appears.

4) Observe, save, and/or print results.

a) Preview the results to assure they appear as desired.

b) Save the file to an appropriate location, if desired.

c) To print the results, navigate to File > Print and print all worksheets flipping the page on the short side.

Future Improvements

There are several improvements that can be made for this document, and many have been suggested to work on for next quarter. These are minor changes to the Excel layout, including adding an operation number column, making tool changes distinct, decreasing row height to fit on one line, adding a column for stick out, moving units into the header row, and center justifying the text. For checking purposes, conditional formatting may be applied to the Excel sheet to highlight worrisome operations or parameter measurements. On a larger scale, improvements can be made to allow for greater customization. A prompt could be added to select or remove wanted parameters and optional parameters. Another question could ask for a tool list or select specific programs to gather information from. Additionally, some more user-friendly warnings and descriptions could be added or improved on current ones. The macro was created to be adaptable and scalable to add or remove parameters and support multiple setups.

Conclusion

From day one, this macro project felt reasonable, though I had some doubts about getting it done along the way. As such, I would only recommend starting similar projects to ambitious students with a fair programming background. Since it was my first time doing any Visual Basics work and learning about the background of the CATIA structure, a steep learning curve had to be tackled to make things work. Therefore, a bulk of the work was sifting through documentation and relevant examples. A major additional challenge was there was incredibly little or no examples for some of the problems encountered. Thus, most issues encountered had to be solved through finding similar problems that were documented and achieved or a three step approach of (1) find how or where it would be solved without macros in CATIA, (2) find out the corresponding location using debugging tools, (3) then implement programmatically. I believe the critical element to my success was due to the CATIA documentation and example macros provided as well as any assistance provided my advisor. Overall, I am very satisfied with my accomplishments and learned a lot in the process through my perseverance.