HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

GRADUATION RESEARCH 2

Excel Add-ins and Functions

PHAN DINH TUONG
Tuong.pd164582@sis.hust.edu.vn

School of Information and Communication Technology
Global ICT Program

Supervisor:	MSc. Nguyen Duc Tien	______________________ Signature of Supervisor
Department:	Computer Engineering	
School:	Information and Communication Technology	

HANOI, 12/2020

RESEARCH TOPIC

Write Excel add-ins and functions with custom functionalities, compile to dynamic link library (.dll and .xll files) to run in Excel spreadsheet.

Supervisor
Signature and full name

Acknowledgement

For the second time, School of Information and Communication Technology, Hanoi University of Science and Technology, created another chance, the Graduation Research 2, for students to research on the interesting aspect of their study field. I truly appreciate that.

I would like to thank my supervisor, MSc. Nguyen Duc Tien, Department of Computer Engineering, School of Information and Communication Technology, Hanoi University of Science and Technology, for his guidance, although it is the mid-night, or his busy day with other works.

Once again, wish you all a happy and successful life!

About research

The research inludes 2 projects, ExcelAddins and ExcelFunction. The two projects provide the specific real problem of Exel add-ins and functions. The details are in the main content.

These two projects are written in C# programming language, IDE Microsoft Visual Studio Community 2019 (Version 16.7.5), with NuGet Package Manager, ResXManager, Github to control source code remotely.

Student

Signature and full name

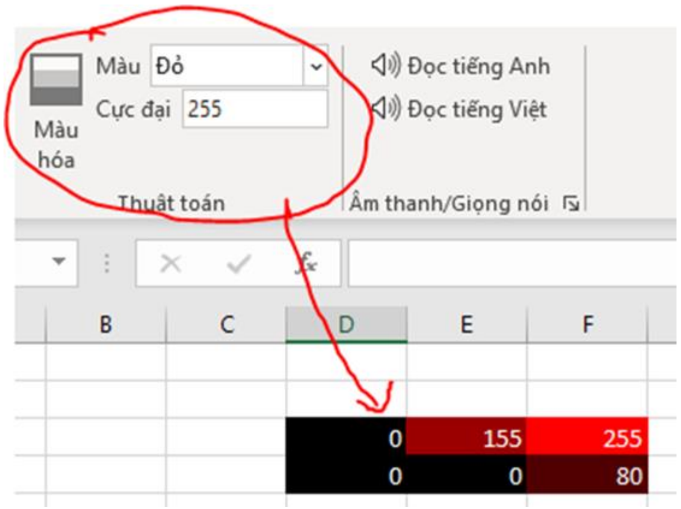
TABLE OF CONTENTS

CHAPTER 1. **MAIN PROGRESS**

1.1 Cell colorization

1.1.1 About

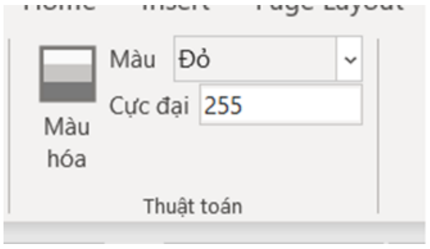
- This feature belongs to ExcelAddIn project.
- User chooses a range of cells, with selected color (red, green, blue) and saturation.
- User clicks on "Colorize".
- The chosen cells will be colorized with corresponding color and saturation.



- Update: Add gray to color, change font color based on cell color (if saturation > 128, font color is black, otherwise, the font color is white), keep input text.

1.1.2 Detail

- In group Algorithm (Thuật toán) on Add-ins tab, there are a button Colorize (Màu hóa), drop down ColorRGB (Màu), edit box editSaturationPeak (Cực đại) as the figure:



- In MyRibon.Designer class, buttonColorize_Click is invoked whenever user click **Colorize** in MyRibon class with arguments color and saturation:

```
this.buttonColorize.Click          +=          (sender,e)          =>
this.buttonColorize_Click(dropDownColorRGB.SelectedItem.OfficeImage
Id,editSaturationPeak.Text);
```

  - In method buttonColorize_Click in MyRibon class, the cells are selected:

```
//get selected cells
```

```
   Range currentRange = (Range)Globals.ThisAddIn.Application.Selection
as
                    Microsoft.Office.Interop.Excel.Range;
   if (currentRange == null) return;
```

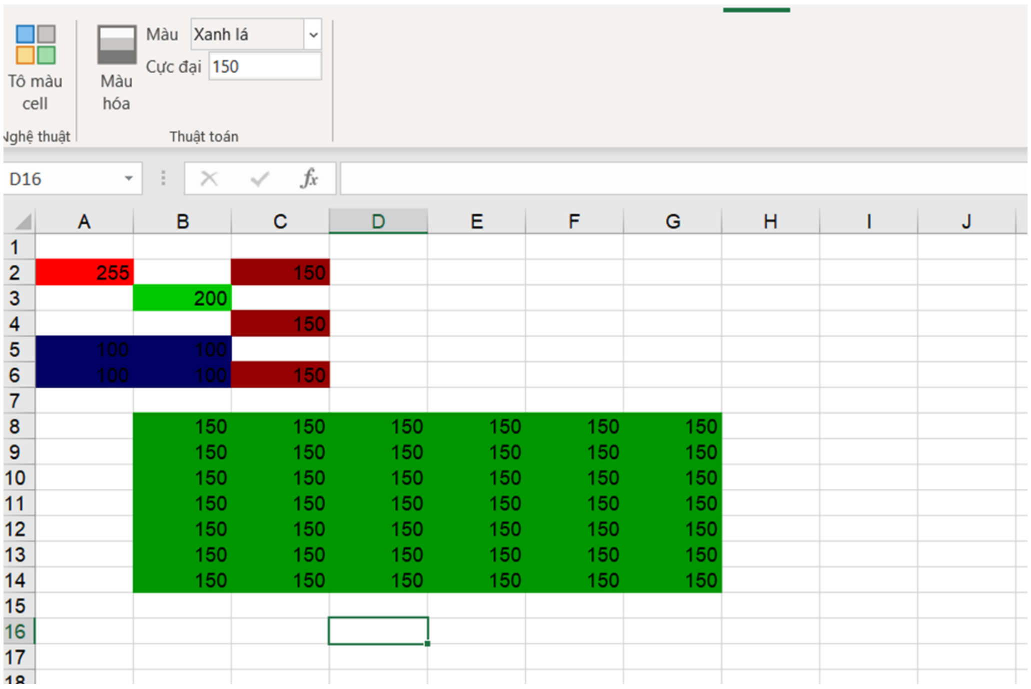
- The saturation value is checked if it is valid:

```
int saturationInt; //saturation value in integer
bool isNumeric = int.TryParse(saturation,out saturationInt); //boolean
to check if saturation is a valid number
if (isNumeric) //check if the saturation is a number
if (saturationInt >= 0 && saturationInt <= 255) //check if the
saturation value is in range 0 to 255
```

- If valid, the color to colorize is checked and the saturation value is assigned to the cells:

```
switch (color) // color to display with saturation
      {
         case "AppointmentColor1": currentRange.Interior.Color =
Color.FromArgb(saturationInt, 0, 0); break; //red
         case "AppointmentColor2": currentRange.Interior.Color =
Color.FromArgb(0, 0, saturationInt); break; //green
         case "AppointmentColor3": currentRange.Interior.Color =
Color.FromArgb(0, saturationInt, 0); break; //blue
         default: break;
      }
currentRange.Value = saturationInt; // cell value = saturation value
```

- Example:



Màu: Xanh lá
Cực đại: 150

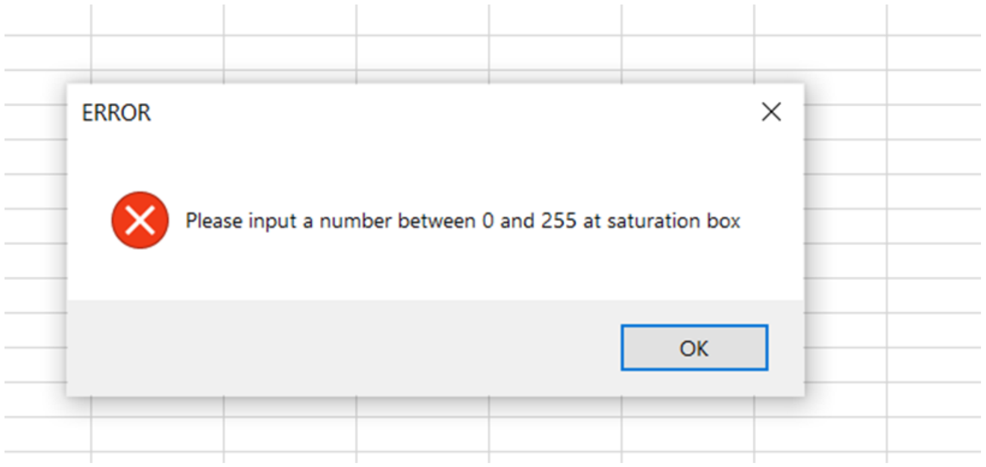
	A	B	C	D	E	F	G	H	I	J
1										
2	255		150							
3		200								
4			150							
5	100	100								
6	100	100	150							
7										
8		150	150	150	150	150	150			
9		150	150	150	150	150	150			
10		150	150	150	150	150	150			
11		150	150	150	150	150	150			
12		150	150	150	150	150	150			
13		150	150	150	150	150	150			
14		150	150	150	150	150	150			
15										
16										
17										

- If invalid, the error messages will be displayed:

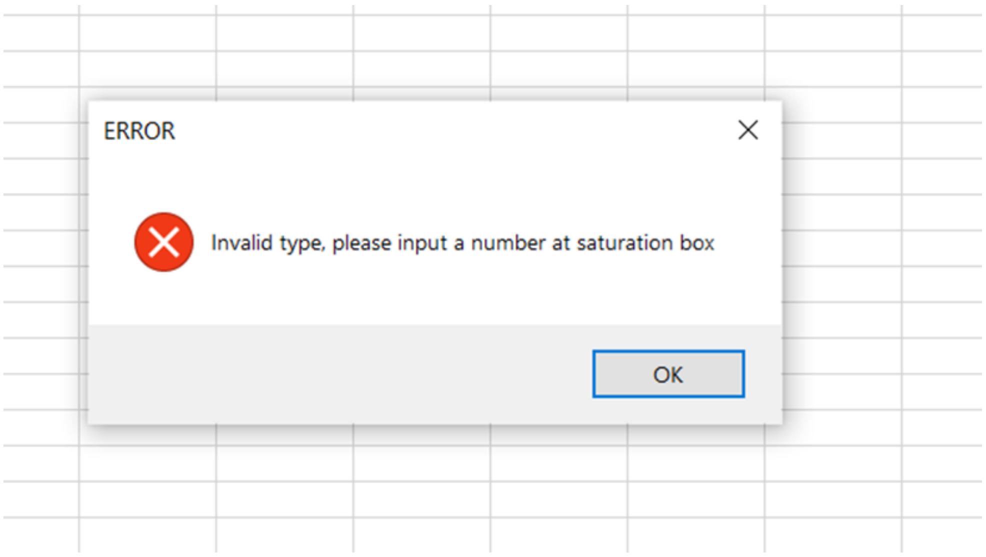
```
//saturation is not in range 0 to 255
```

```
MessageBox.Show("Please input a number between 0 and 255 at saturation
box", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
//saturation is not a valid number
MessageBox.Show("Invalid type, please input a number at saturation
box", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
```

- Example:


ERROR

Please input a number between 0 and 255 at saturation box


ERROR

Invalid type, please input a number at saturation box

1.2 QRCode generator function

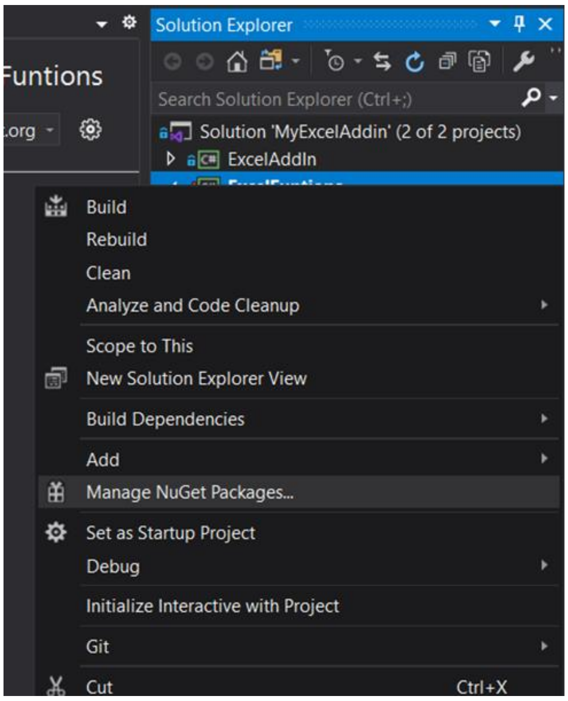
1.2.1 About

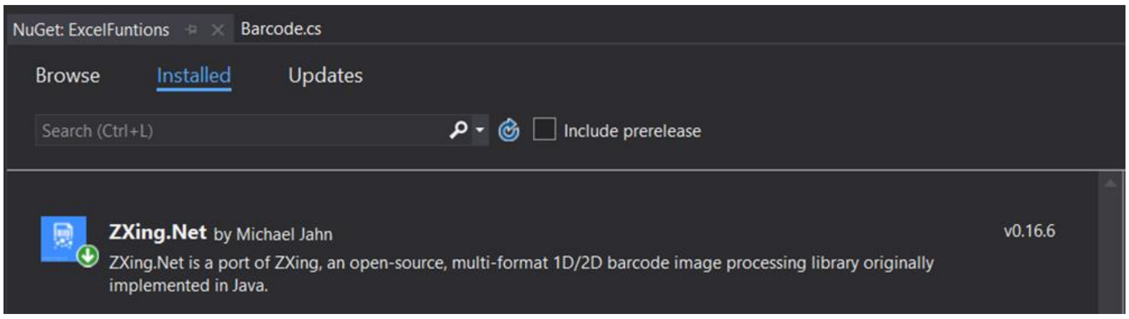
- This feature belongs to ExcelFunctions project.
- Using ZXing library (https://github.com/zxing/zxing) (instead of Google API) to generate QRCode image by input text.

1.2.2 Detail

- ZXing library can be installed on Visual Studio as NuGet package:

  On Solution Explorer, right click on one project, choose "Manage NuGet Packages…", browse and install "ZXing.NET":





ZXing.Net by Michael Jahn

ZXing.Net is a port of ZXing, an open-source, multi-format 1D/2D barcode image processing library originally implemented in Java.

- In Barcode class, an excel function QRCodeZ (Z stands for ZXing), with a string Text parameter, was created:

```
//Generate QRCode using ZXing library
        [ExcelDna.Integration.ExcelFunction(Description = "QRCode generator by ZXing lib")]
        public static object QRCodeZ(
            [ExcelDna.Integration.ExcelArgument(Description = "Text to be transformed to QRCode. Example: \"hello\"")]
            string Text)
```

- The Text string is read by ZXing library. Next step, it is transformed into QRCode image, saved to a file, then loaded to the current excel file.
- Update (12/10/2020): not necessary to save the QRCode image to a file, just save it to clipboard and paste to the worksheet.
- Update (23/10/2020): Add a parameter `int option` to the function QRCode, with option 1 to use ZXing library to generate QRCode image, option 2 to use Google API. The ZXing option uses Path.GetTempFileName() to save the QRCode image to local storage.

1.3 KipThi function

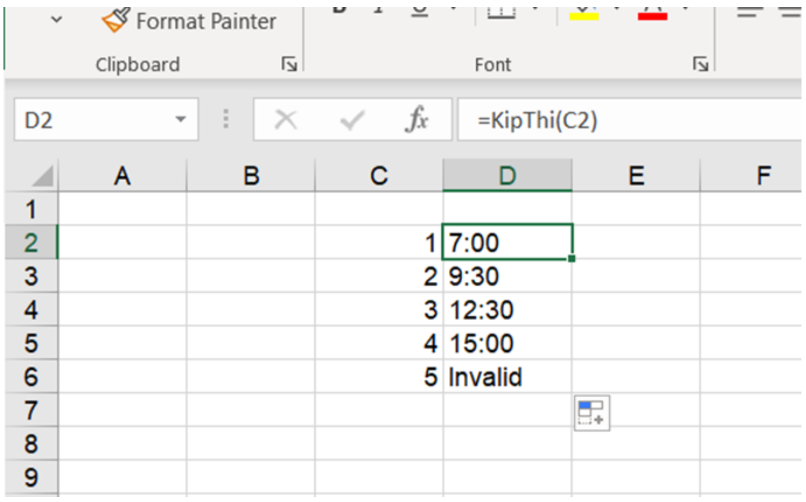
1.3.1 About

- This feature belongs to ExcelFunctions project.

- Function Kipthi(STT) to show the starting time of the exam:
    + KipThi(1) -> 07:00
    + KipThi(2) -> 09:30
    + KipThi(3) -> 12:30
    + KipThi(4) -> 15:00

1.3.2 Detail

- In Hust class, a function KipThi was added with an int Kip parameter. The function will return the corresponding starting time of the exam:



=KipThi(C2)

	A	B	C	D	E	F
1						
2			1	7:00		
3			2	9:30		
4			3	12:30		
5			4	15:00		
6			5	Invalid		
7						
8						
9						

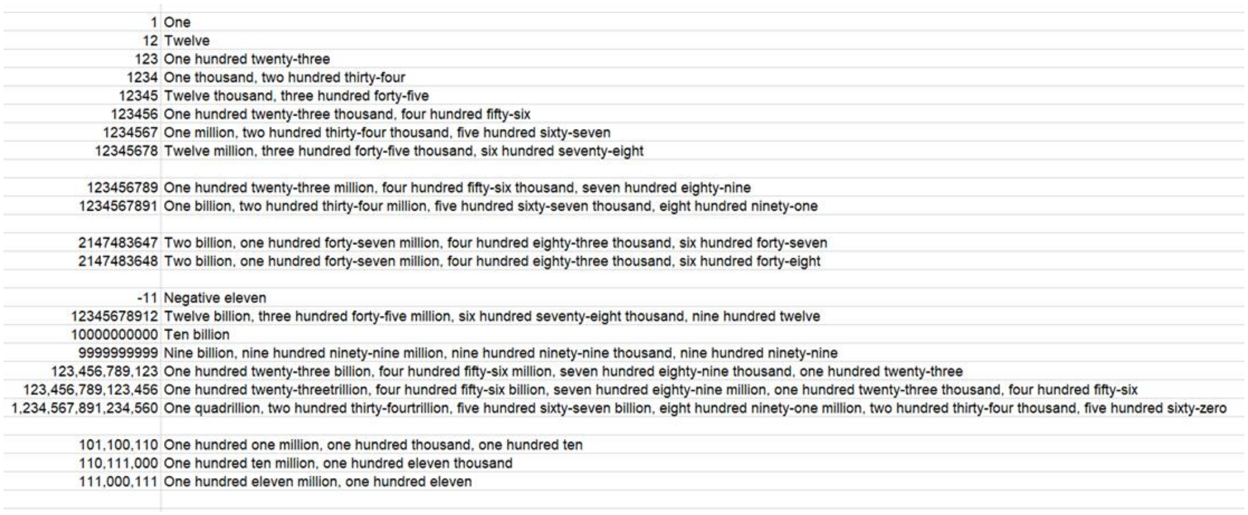
1.4 InText function

1.4.1 About

- This feature belongs to ExcelFunctions project.
- Function InText will transform number into text in English.

1.4.2 Detail

- In FinanceTools class, a function InText was created with parameter `int number`, which will return the text value of the number in English.
- Another function ReadMoneyInText to divide the original number into smaller 3-digit numbers, and a function Read3DigitNumber to tranform these 3-digit numbers into text.
- TODO: lỗi tiếng việt, vd: 1001 đọc là "một nghìn không trăm linh một". Done (check thêm ở trước có số không).
- Example:



1	One
12	Twelve
123	One hundred twenty-three
1234	One thousand, two hundred thirty-four
12345	Twelve thousand, three hundred forty-five
123456	One hundred twenty-three thousand, four hundred fifty-six
1234567	One million, two hundred thirty-four thousand, five hundred sixty-seven
12345678	Twelve million, three hundred forty-five thousand, six hundred seventy-eight
123456789	One hundred twenty-three million, four hundred fifty-six thousand, seven hundred eighty-nine
1234567891	One billion, two hundred thirty-four million, five hundred sixty-seven thousand, eight hundred ninety-one
2147483647	Two billion, one hundred forty-seven million, four hundred eighty-three thousand, six hundred forty-seven
2147483648	Two billion, one hundred forty-seven million, four hundred eighty-three thousand, six hundred forty-eight
-11	Negative eleven
12345678912	Twelve billion, three hundred forty-five million, six hundred seventy-eight thousand, nine hundred twelve
10000000000	Ten billion
9999999999	Nine billion, nine hundred ninety-nine million, nine hundred ninety-nine thousand, nine hundred ninety-nine
123,456,789,123	One hundred twenty-three billion, four hundred fifty-six million, seven hundred eighty-nine thousand, one hundred twenty-three
123,456,789,123,456	One hundred twenty-threetrillion, four hundred fifty-six billion, seven hundred eighty-nine million, one hundred twenty-three thousand, four hundred fifty-six
1,234,567,891,234,560	One quadrillion, two hundred thirty-fourtrillion, five hundred sixty-seven billion, eight hundred ninety-one million, two hundred thirty-four thousand, five hundred sixty-zero
101,100,110	One hundred one million, one hundred thousand, one hundred ten
110,111,000	One hundred ten million, one hundred eleven thousand
111,000,111	One hundred eleven million, one hundred eleven

1.5 StringCellFormatter and StringFontFormatter functions

1.5.1 About

- These features belong to ExcelFunction project.
- Function StringCellFormatter(text, CellBackgroundColor) will return the input text and change the background color of the cell.
- Function StringFontFormatter(text, FontName, FontSize, FontColor) will return the input text and change font type, font size, font color (type #3456FC for RGB).

1.5.2 Detail

- The functions cannot access to the required properties. Therefore, these functions were canceled.

1.6 Cortana voice recognition

1.6.1 About

- This feature belongs to ExcelAddIn project.
- There is a button on ribbon bar, when user clicks that button, Cortana will return the text from voice recognition to a cell.

1.6.2 Detail

- Added a button on ribbon bar.
- Used System.Speech.Recognition
- Recognized but not so accurate.
- Show progress of listening on the cell.
- This add-in is for research only because there is a better solution for voice recognition.

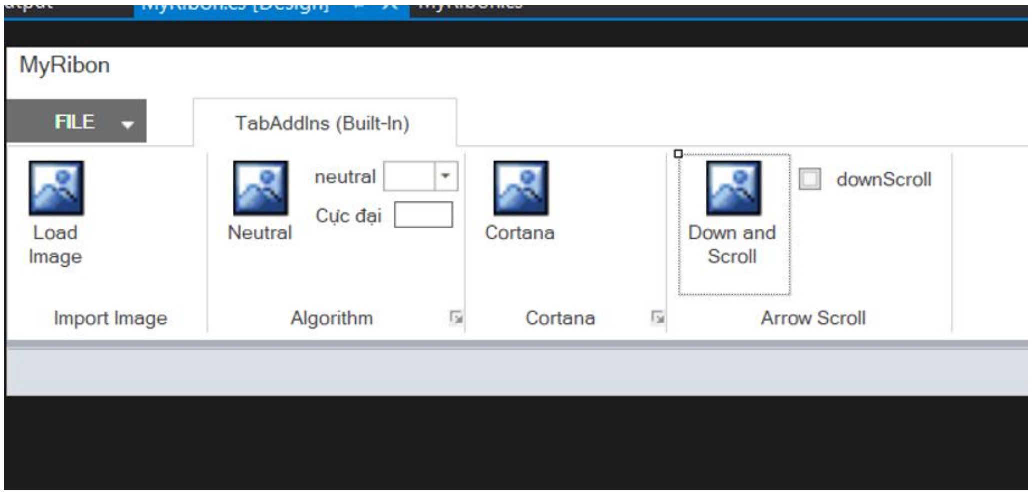
1.7 ArrowMouseScroll

1.7.1 About

- This feature belongs to ExcelAddIn project.
- When user presses down arrow, the workbook also scrolls.

1.7.2 Detail

- A new group Arrow Scroll and a button Down and Scroll were created:

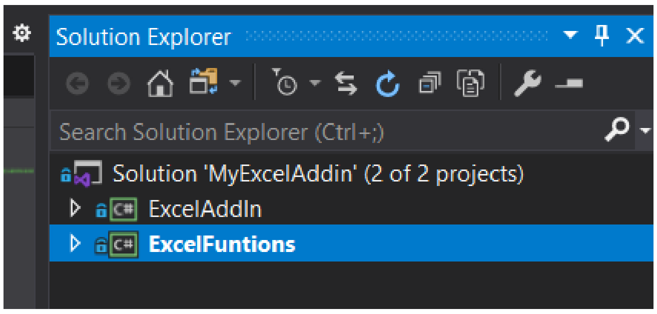


- In this button’s click event, this snippet of code was added:

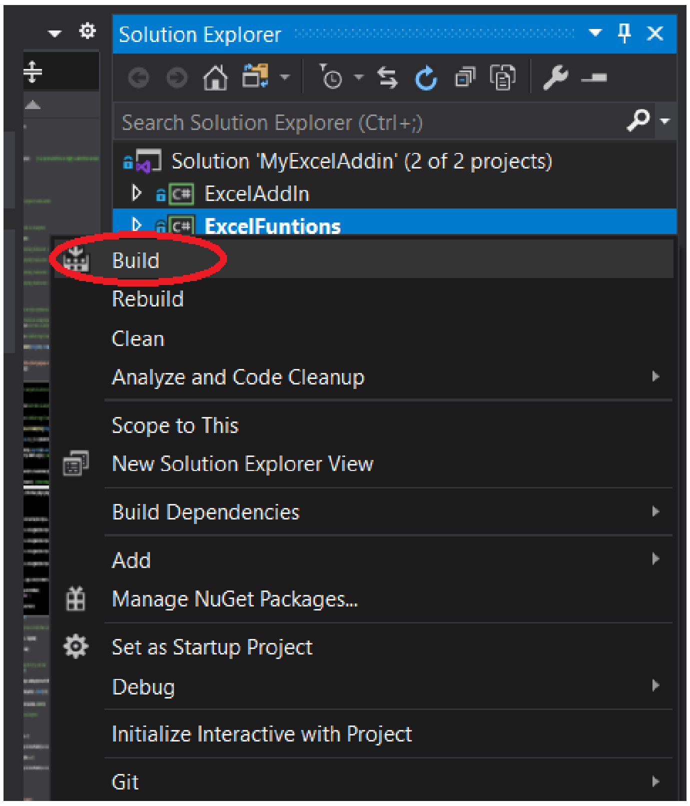
```
SendKeys.Send("{Down}{SCROLLLOCK}{Down}");
```

- It will send the keys in the brackets, including 3 keys: Down (down arrow), SCROLLLOCK (Scroll Lock), and another Down.
- The first Down will make the pointer down 1 cell, scroll lock and the second down will make the workbook scroll one row.

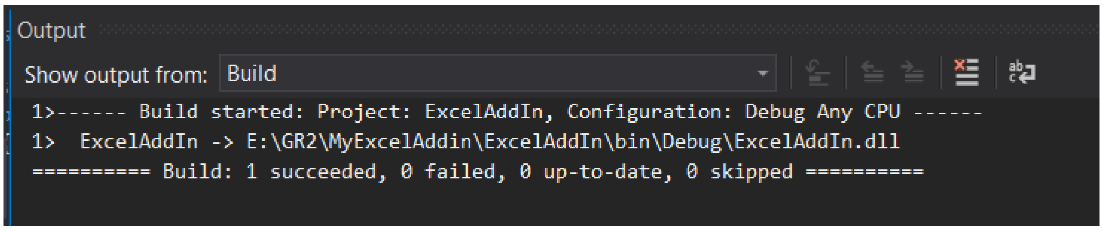
- Clone the source code, open it in Visual Studio. In the Solution Explorer, there are 2 projects, ExcelAddIn and ExcelFunctions:


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'MyExcelAddin' (2 of 2 projects)
ExcelAddIn
ExcelFuntions

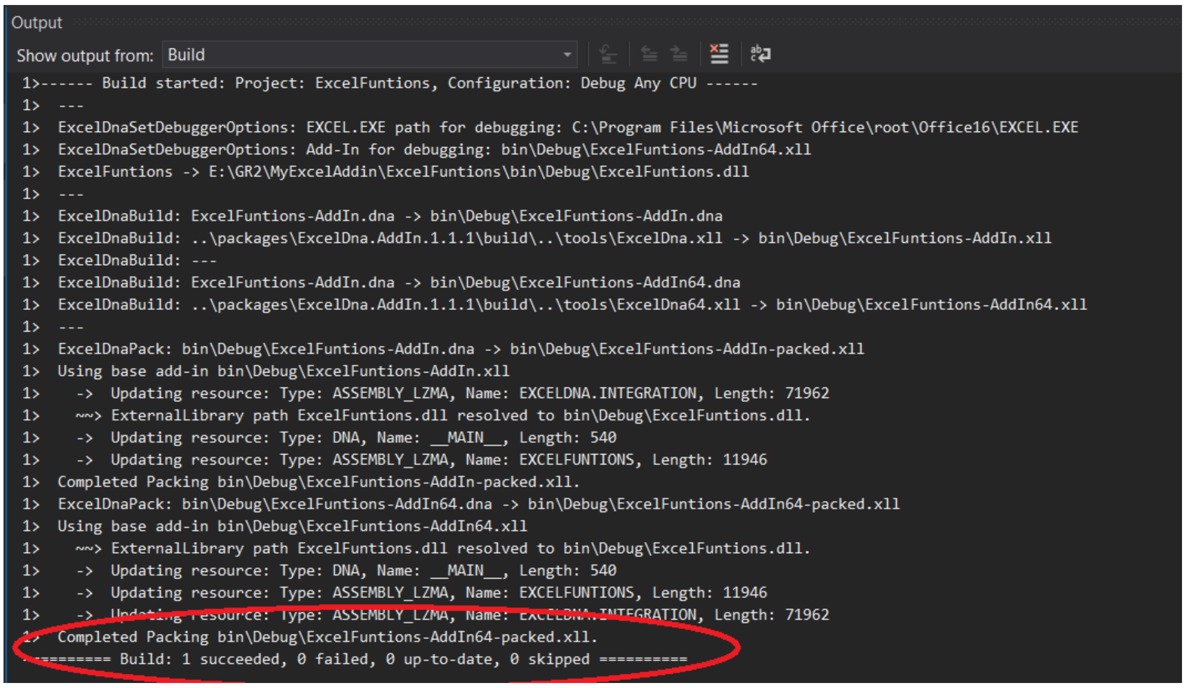
- Right click to one project and choose Build to build and run it:



- See the result after building the project, pay attention the path of the result file:

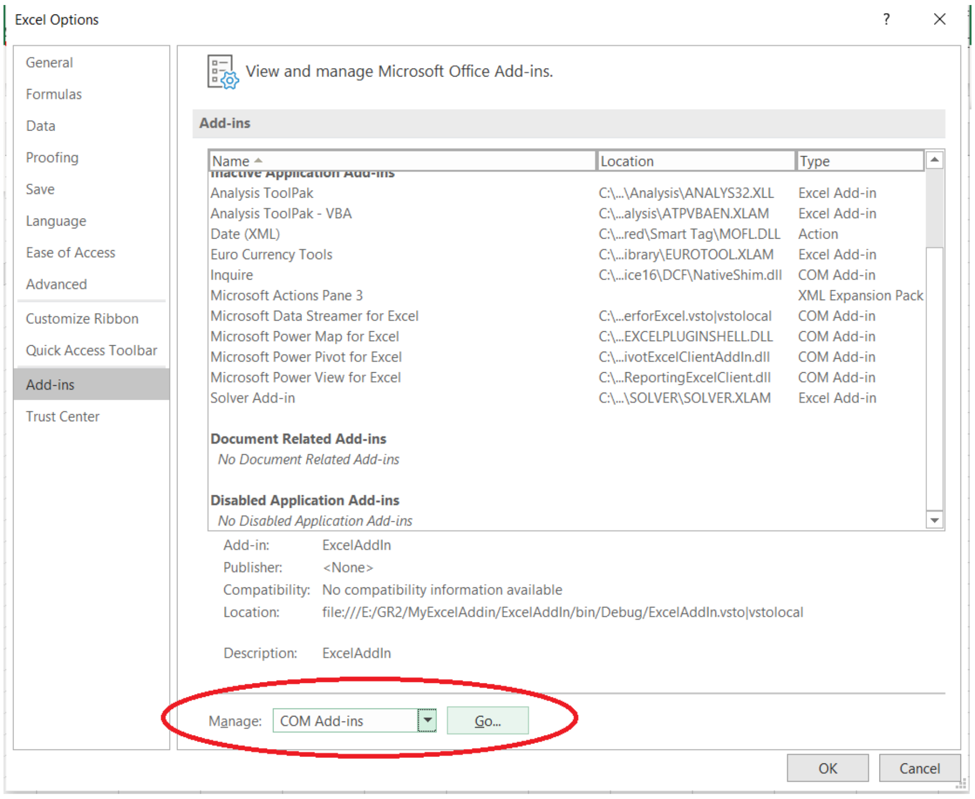


```
1>------ Build started: Project: ExcelAddIn, Configuration: Debug Any CPU ------
1>  ExcelAddIn -> E:\GR2\MyExcelAddin\ExcelAddIn\bin\Debug\ExcelAddIn.dll
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```



```
Output
Show output from: Build
1>------ Build started: Project: ExcelFuntions, Configuration: Debug Any CPU ------
1>  ---
1>  ExcelDnaSetDebuggerOptions: EXCEL.EXE path for debugging: C:\Program Files\Microsoft Office\root\Office16\EXCEL.EXE
1>  ExcelDnaSetDebuggerOptions: Add-In for debugging: bin\Debug\ExcelFuntions-AddIn64.xll
1>  ExcelFuntions -> E:\GR2\MyExcelAddin\ExcelFuntions\bin\Debug\ExcelFuntions.dll
1>  ---
1>  ExcelDnaBuild: ExcelFuntions-AddIn.dna -> bin\Debug\ExcelFuntions-AddIn.dna
1>  ExcelDnaBuild: ..\packages\ExcelDna.AddIn.1.1.1\build\..\tools\ExcelDna.xll -> bin\Debug\ExcelFuntions-AddIn.xll
1>  ExcelDnaBuild: ---
1>  ExcelDnaBuild: ExcelFuntions-AddIn.dna -> bin\Debug\ExcelFuntions-AddIn64.dna
1>  ExcelDnaBuild: ..\packages\ExcelDna.AddIn.1.1.1\build\..\tools\ExcelDna64.xll -> bin\Debug\ExcelFuntions-AddIn64.xll
1>  ---
1>  ExcelDnaPack: bin\Debug\ExcelFuntions-AddIn.dna -> bin\Debug\ExcelFuntions-AddIn-packed.xll
1>  Using base add-in bin\Debug\ExcelFuntions-AddIn.xll
1>    -> Updating resource: Type: ASSEMBLY_LZMA, Name: EXCELDNA.INTEGRATION, Length: 71962
1>    ~~> ExternalLibrary path ExcelFuntions.dll resolved to bin\Debug\ExcelFuntions.dll.
1>    -> Updating resource: Type: DNA, Name: __MAIN__, Length: 540
1>    -> Updating resource: Type: ASSEMBLY_LZMA, Name: EXCELFUNTIONS, Length: 11946
1>  Completed Packing bin\Debug\ExcelFuntions-AddIn-packed.xll.
1>  ExcelDnaPack: bin\Debug\ExcelFuntions-AddIn64.dna -> bin\Debug\ExcelFuntions-AddIn64-packed.xll
1>  Using base add-in bin\Debug\ExcelFuntions-AddIn64.xll
1>    ~~> ExternalLibrary path ExcelFuntions.dll resolved to bin\Debug\ExcelFuntions.dll.
1>    -> Updating resource: Type: DNA, Name: __MAIN__, Length: 540
1>    -> Updating resource: Type: ASSEMBLY_LZMA, Name: EXCELFUNTIONS, Length: 11946
1>    -> Updating resource: Type: ASSEMBLY_LZMA, Name: EXCELDNA.INTEGRATION, Length: 71962
1>  Completed Packing bin\Debug\ExcelFuntions-AddIn64-packed.xll.
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

- To run ExcelAddIn project, open one Excel workbook, on the ribbon bar, choose File -> Options, the Excel Options window will appear, in this window, choose Add-ins -> Manage, choose COM Add-ins, and Go:



View and manage Microsoft Office Add-ins.

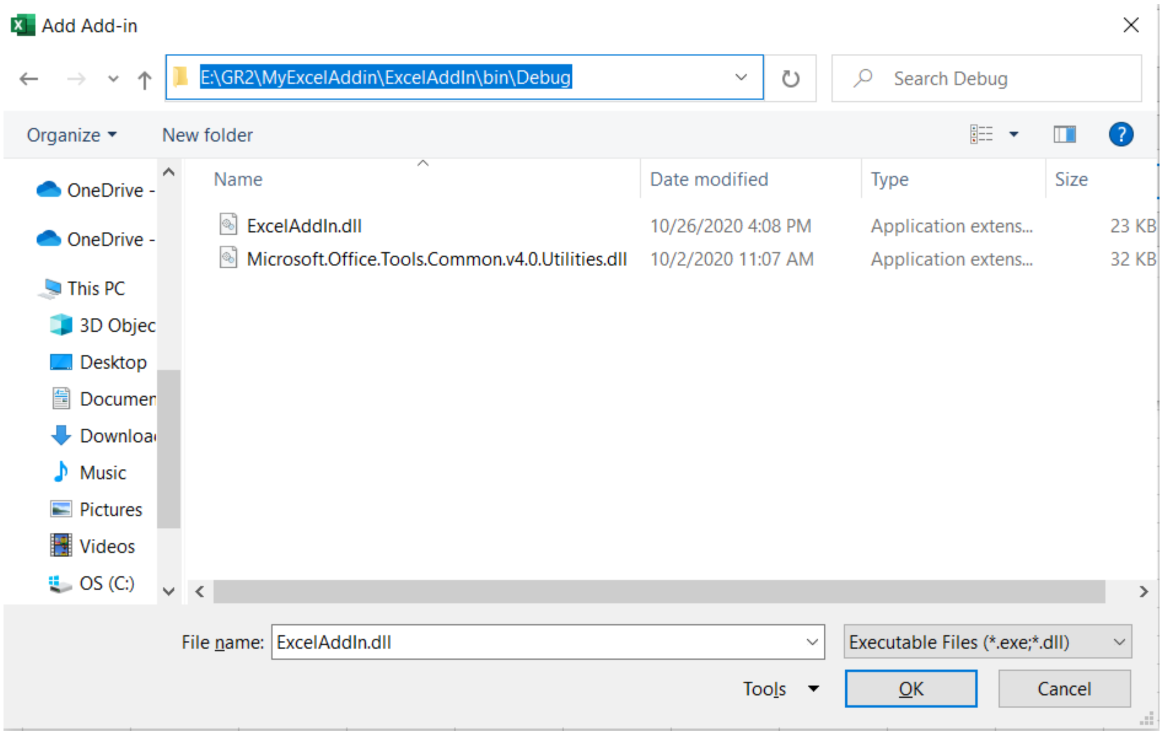
Add-ins

Name	Location	Type
Inactive Application Add-ins		
Analysis ToolPak	C:\...\Analysis\ANALYS32.XLL	Excel Add-in
Analysis ToolPak - VBA	C:\...alysis\ATPVBAEN.XLAM	Excel Add-in
Date (XML)	C:\...red\Smart Tag\MOFL.DLL	Action
Euro Currency Tools	C:\...ibrary\EUROTOOL.XLAM	Excel Add-in
Inquire	C:\...ice16\DCF\NativeShim.dll	COM Add-in
Microsoft Actions Pane 3		XML Expansion Pack
Microsoft Data Streamer for Excel	C:\...erforExcel.vsto|vstolocal	COM Add-in
Microsoft Power Map for Excel	C:\...EXCELPLUGINSHELL.DLL	COM Add-in
Microsoft Power Pivot for Excel	C:\...ivotExcelClientAddIn.dll	COM Add-in
Microsoft Power View for Excel	C:\...ReportingExcelClient.dll	COM Add-in
Solver Add-in	C:\...\SOLVER\SOLVER.XLAM	Excel Add-in
Document Related Add-ins		
No Document Related Add-ins		
Disabled Application Add-ins		
No Disabled Application Add-ins		

Add-in:	ExcelAddIn
Publisher:	<None>
Compatibility:	No compatibility information available
Location:	file:///E:/GR2/MyExcelAddin/ExcelAddIn/bin/Debug/ExcelAddIn.vsto|vstolocal
Description:	ExcelAddIn

Manage: COM Add-ins Go...

- Add the Dynamic Link Library (.dll file) to the COM Add-ins windows, the path of this file is the build result of the ExcelAddIn project:

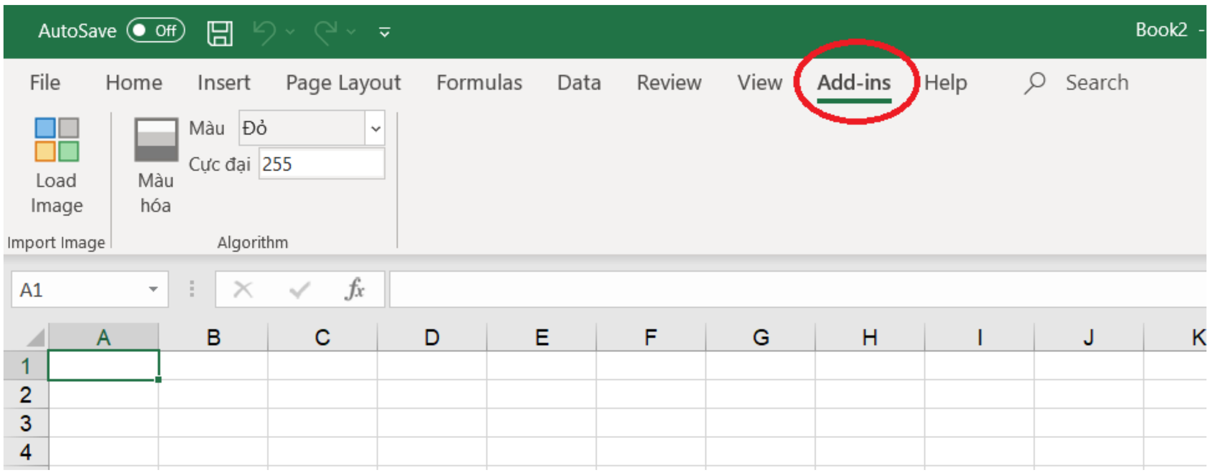


Name	Date modified	Type	Size
ExcelAddIn.dll	10/26/2020 4:08 PM	Application extens...	23 KB
Microsoft.Office.Tools.Common.v4.0.Utilities.dll	10/2/2020 11:07 AM	Application extens...	32 KB

File name: ExcelAddIn.dll

Executable Files (*.exe;*.dll)

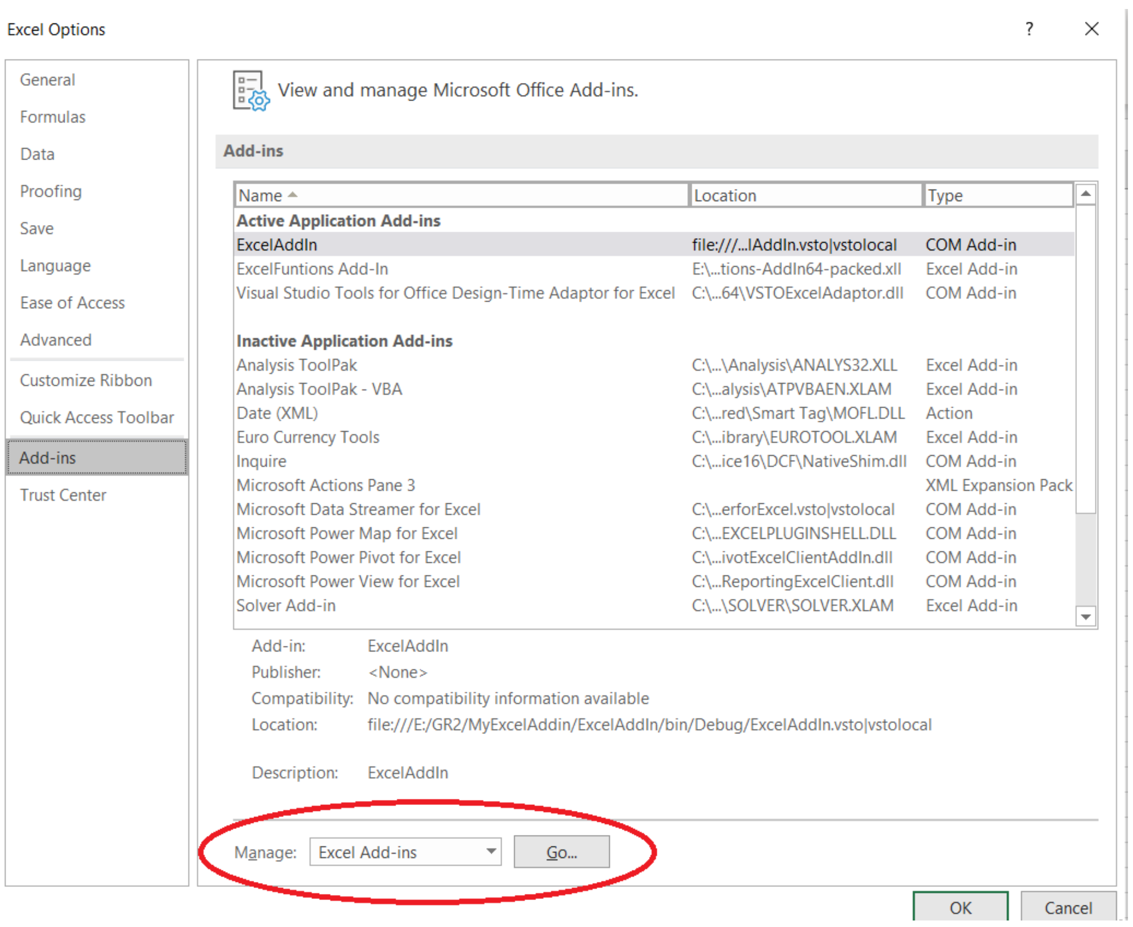
- Click OK, on the current workbook, the new tab Add-ins will appear on the ribbon bar(if not, you may have to restart the Excel workbook):



Màu: Đỏ

Cực đại: 255

- To run ExcelFunctions project, on the Manage section of the Excel Options window, choose Excel Add-ins and Go:



Excel Options

General
Formulas
Data
Proofing
Save
Language
Ease of Access
Advanced
Customize Ribbon
Quick Access Toolbar
Add-ins
Trust Center

View and manage Microsoft Office Add-ins.

Add-ins

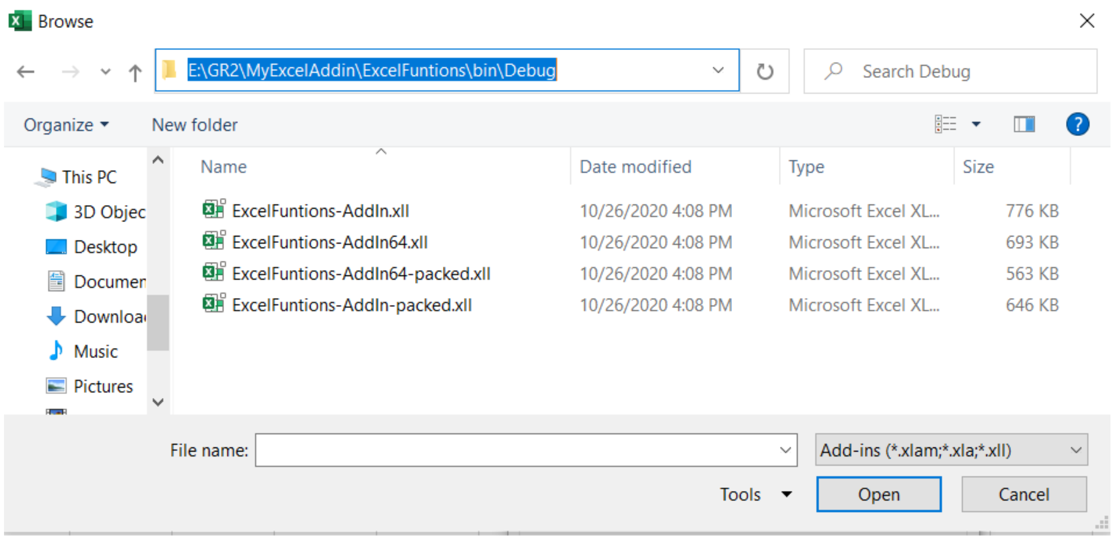
Name	Location	Type
Active Application Add-ins		
ExcelAddIn	file:///...lAddIn.vsto|vstolocal	COM Add-in
ExcelFuntions Add-In	E:\...tions-AddIn64-packed.xll	Excel Add-in
Visual Studio Tools for Office Design-Time Adaptor for Excel	C:\...64\VSTOExcelAdaptor.dll	COM Add-in
Inactive Application Add-ins		
Analysis ToolPak	C:\...\Analysis\ANALYS32.XLL	Excel Add-in
Analysis ToolPak - VBA	C:\...alysis\ATPVBAEN.XLAM	Excel Add-in
Date (XML)	C:\...red\Smart Tag\MOFL.DLL	Action
Euro Currency Tools	C:\...ibrary\EUROTOOL.XLAM	Excel Add-in
Inquire	C:\...ice16\DCF\NativeShim.dll	COM Add-in
Microsoft Actions Pane 3		XML Expansion Pack
Microsoft Data Streamer for Excel	C:\...erforExcel.vsto|vstolocal	COM Add-in
Microsoft Power Map for Excel	C:\...EXCELPLUGINSHELL.DLL	COM Add-in
Microsoft Power Pivot for Excel	C:\...ivotExcelClientAddIn.dll	COM Add-in
Microsoft Power View for Excel	C:\...ReportingExcelClient.dll	COM Add-in
Solver Add-in	C:\...\SOLVER\SOLVER.XLAM	Excel Add-in

Add-in:	ExcelAddIn
Publisher:	<None>
Compatibility:	No compatibility information available
Location:	file:///E:/GR2/MyExcelAddin/ExcelAddIn/bin/Debug/ExcelAddIn.vsto|vstolocal
Description:	ExcelAddIn

Manage: Excel Add-ins Go...

OK Cancel

- Click Browse button on the new window Add-ins to lead to the path of the build result file (ExcelFunction-AddIn64-packed.xll) of the ExcelFunctions project:



Browse

E:\GR2\MyExcelAddin\ExcelFuntions\bin\Debug

Search Debug

Organize New folder

This PC
3D Objec
Desktop
Documen
Downloa
Music
Pictures

Name	Date modified	Type	Size
ExcelFuntions-AddIn.xll	10/26/2020 4:08 PM	Microsoft Excel XL...	776 KB
ExcelFuntions-AddIn64.xll	10/26/2020 4:08 PM	Microsoft Excel XL...	693 KB
ExcelFuntions-AddIn64-packed.xll	10/26/2020 4:08 PM	Microsoft Excel XL...	563 KB
ExcelFuntions-AddIn-packed.xll	10/26/2020 4:08 PM	Microsoft Excel XL...	646 KB

File name: Add-ins (*.xlam;*.xla;*.xll)

Tools Open Cancel

- Click OK. To test, choose one cell on the workbook, on the Formula Bar, input =QRCode(<option>,”<shape>”,”<string to QRCode>”), for example: =QRCode(1,"shape1","testString").

CHAPTER 3. **CONCLUSION**

The overall result of the projects is successful, though some small parts are not as expected as from the beginning.

These DLL can be used in real cases, and in multiple version of Excel on different Microsoft Windows version.

After this research, I have gained much more knowledge and skills in coding in C#, writing new functions and add-ins for Excel, language in Microsoft Windows management.

REFERENCES

- MSc. Nguyen Duc Tien’s github MyExcelAddin project repository: https://github.com/neittien0110/MyExcelAddin
- This research repository: https://github.com/phandinhtuong/MyExcelAddin
- Microsoft Visual Studio website: https://visualstudio.microsoft.com/
- ZXing library: https://www.nuget.org/packages/ZXing.Net
- ResXManager: https://marketplace.visualstudio.com/items?itemName=TomEnglert.ResXManager
- Windows speech recognition: https://docs.microsoft.com/en-us/windows/uwp/design/input/speech-recognition