

# Week 4 – ABB Robot Teaching Loading Program to Robot

---

Advanced Robotic Systems – MANU2453

Dr Ehsan Asadi, School of Engineering  
RMIT University, Victoria, Australia  
Email: ehsan.asadi@rmit.edu.au

# Content

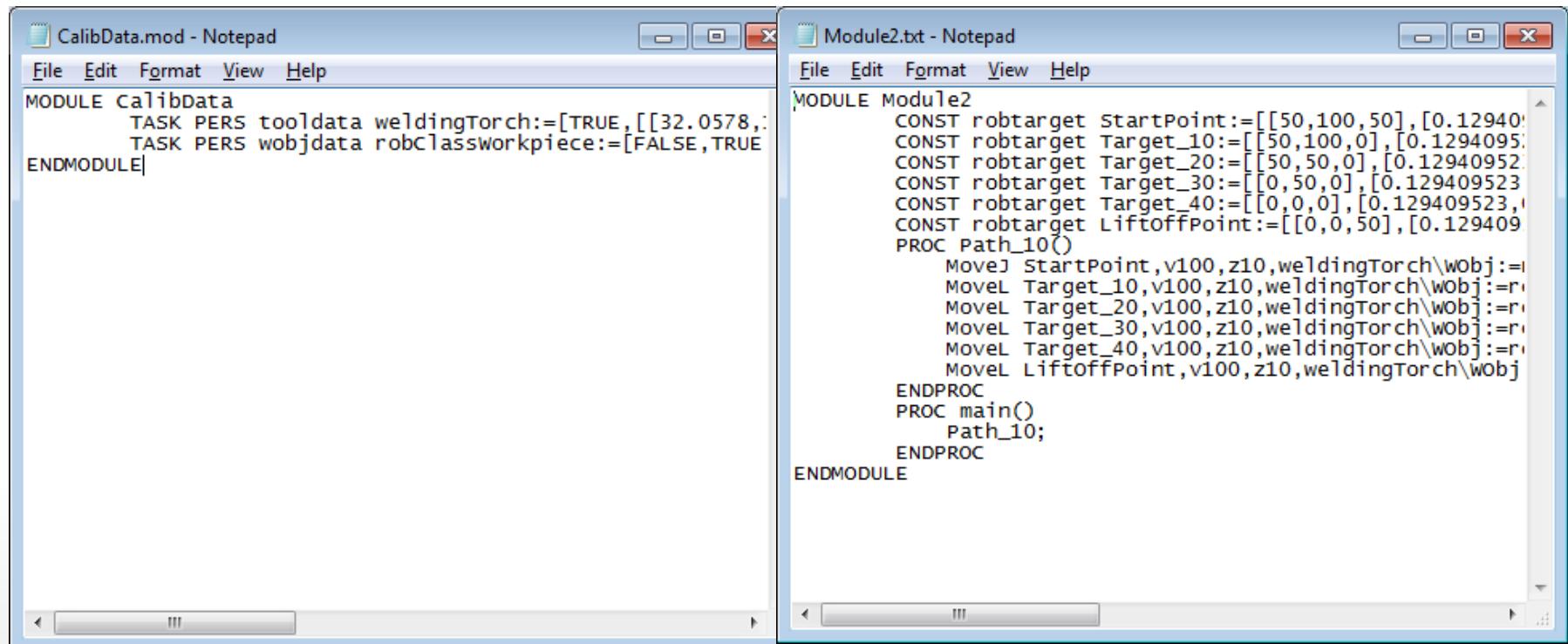
- Introduction
- Backup and Delete Old Program
- Load New Program
- Run New Program
- Delete New Program and Reload Old Program (Optional)

# Content

- Introduction
- Backup and Delete Old Program
- Load New Program
- Run New Program
- Delete New Program and Reload Old Program (Optional)

# Introduction

- In the previous tutorial, we have created two .mod files using **offline programming** method in ABB RobotStudio.
- We have also copied them onto a **USB thumb drive**.



The image shows two side-by-side Notepad windows. The left window is titled 'CalibData.mod - Notepad' and contains the following code:

```
MODULE CalibData
    TASK PERS tooldata weldingTorch:=[TRUE, [[32.0578, :;
    TASK PERS wobjdata robClassWorkpiece:=[FALSE, TRUE
ENDMODULE
```

The right window is titled 'Module2.txt - Notepad' and contains the following code:

```
MODULE Module2
    CONST robtarget StartPoint:=[[50,100,50], [0.1294095];
    CONST robtarget Target_10:=[[50,100,0], [0.1294095];
    CONST robtarget Target_20:=[[50,50,0], [0.12940952];
    CONST robtarget Target_30:=[[0,50,0], [0.129409523];
    CONST robtarget Target_40:=[[0,0,0], [0.129409523];
    CONST robtarget LiftoffPoint:=[[0,0,50], [0.129409523];
    PROC Path_10()
        MoveJ StartPoint,v100,z10,weldingTorch\wobj:=r;
        MoveL Target_10,v100,z10,weldingTorch\wobj:=r;
        MoveL Target_20,v100,z10,weldingTorch\wobj:=r;
        MoveL Target_30,v100,z10,weldingTorch\wobj:=r;
        MoveL Target_40,v100,z10,weldingTorch\wobj:=r;
        MoveL LiftoffPoint,v100,z10,weldingTorch\wobj:=r;
    ENDPROC
    PROC main()
        Path_10;
    ENDPROC
ENDMODULE
```

- We are now ready to **load** them to the **actual robot**.

# Content

- Introduction
- Backup and Delete Old Program
- Load New Program
- Run New Program
- Delete New Program and Reload Old Program (Optional)

# Backup and Delete Old Program

- In the **original program** which is already in the actual robot controller, i.e. “Module1.mod” in this case, there is a **main()** function.
- In the **newly created program**, which is “Module2.mod”, there is also a **main()** function.
- If we load “Module2.mod” onto the robot controller, there will be errors due to **conflict** of the **main()** functions.
- Therefore, we will first **delete** “Module1.mod” before **loading** “Module2.mod”.
- Nevertheless, it is a good idea to **save** “Module1.mod” first so that we can **reload** it later.
- In this section, we will learn how to do this.

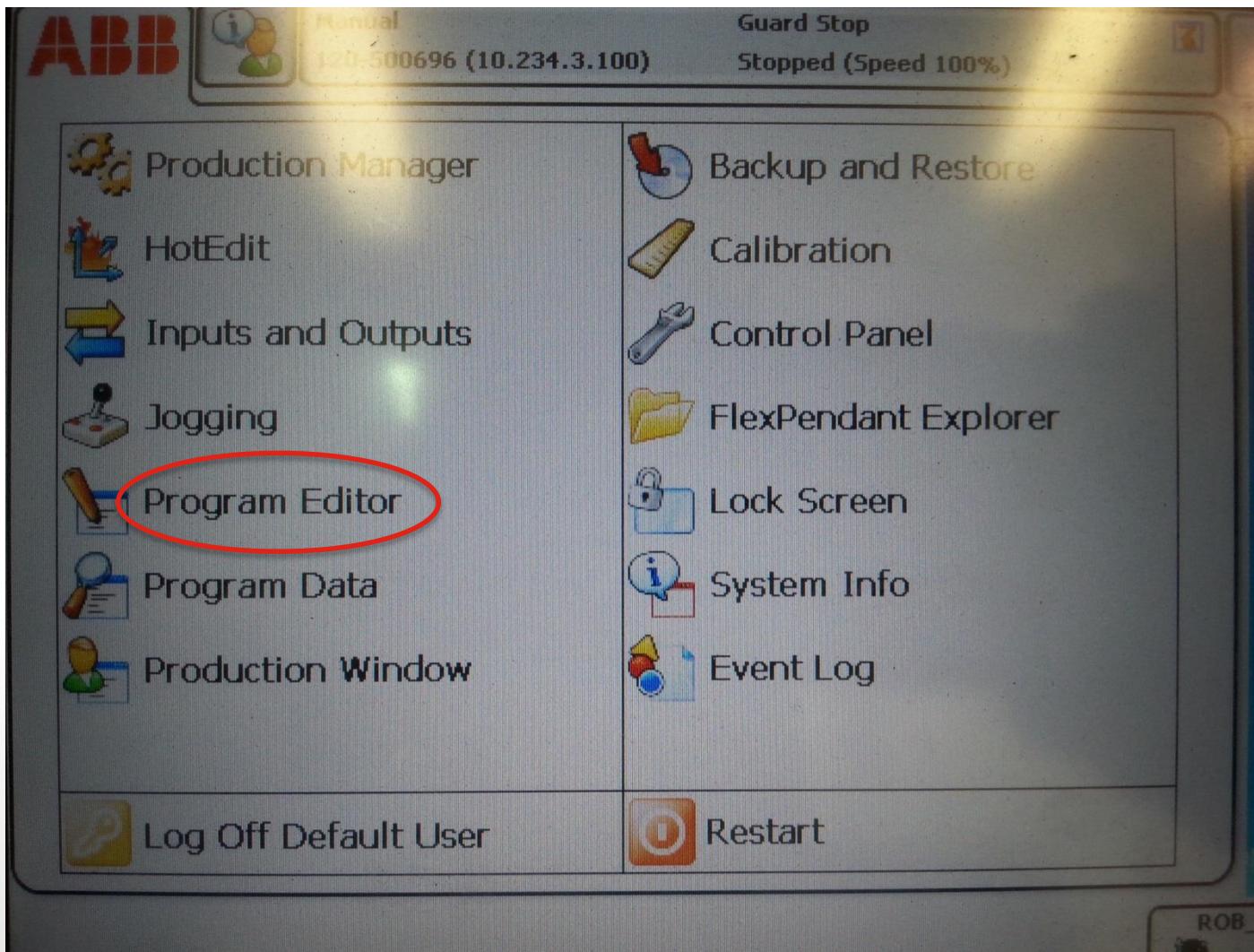
# Backup and Delete Old Program

- First of all, plug the **USB thumb drive** into the **USB port** of the pendant.



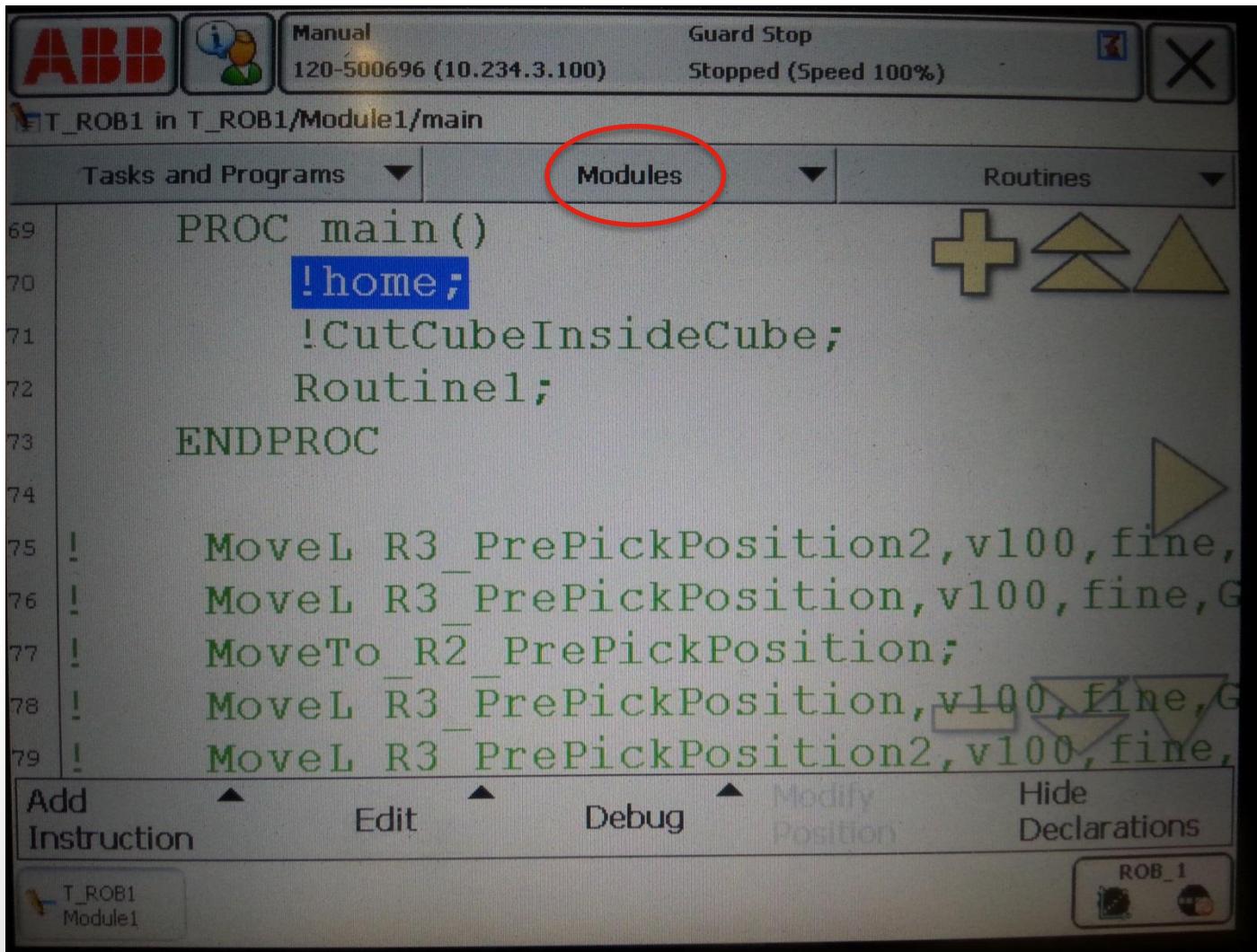
# Backup and Delete Old Program

- On the main screen, choose Program Editor.



# Backup and Delete Old Program

- Click **Modules** on top of the screen.



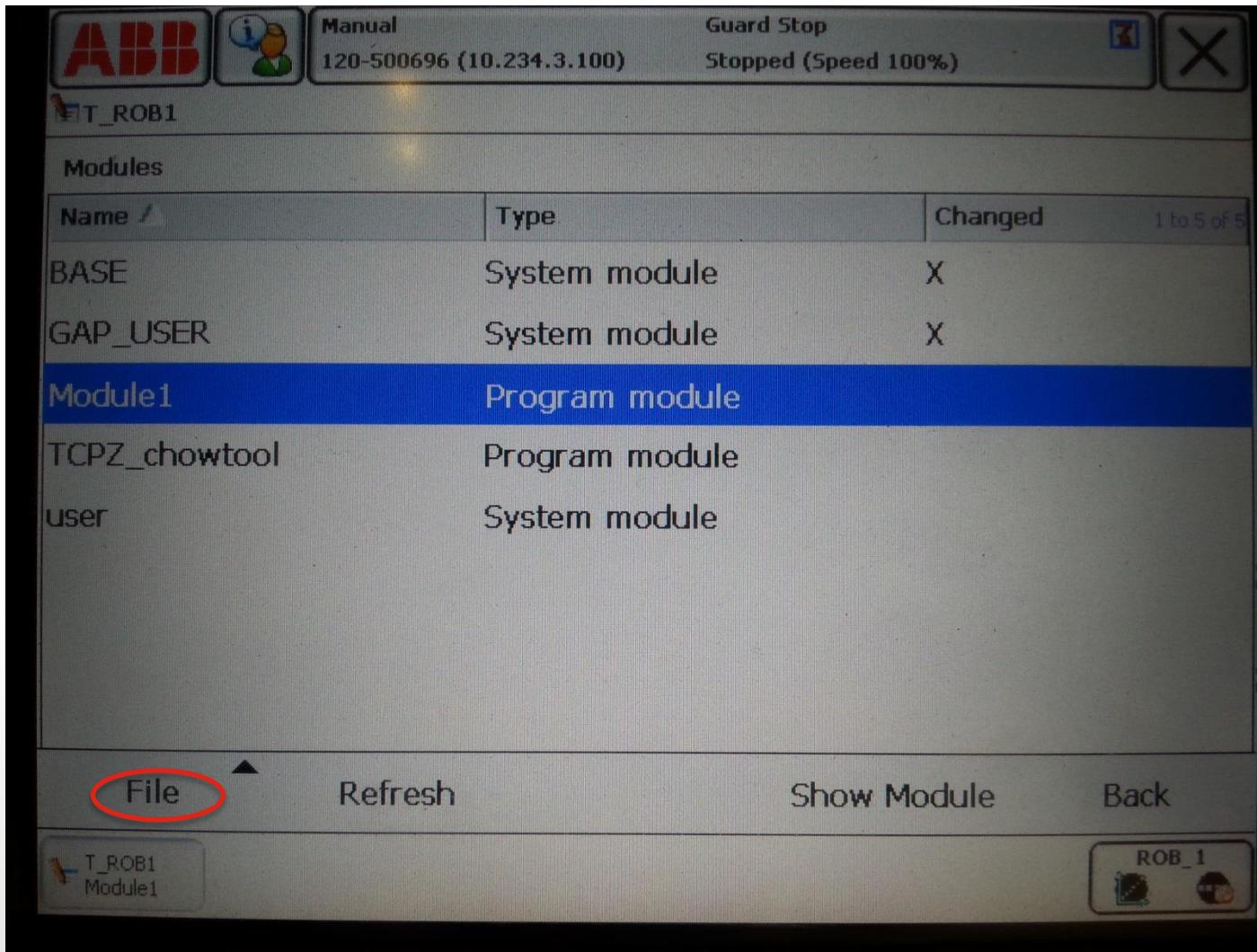
The screenshot shows the ABB R9 software interface. At the top, there is a toolbar with icons for ABB, Manual (IP address 120-500696 (10.234.3.100)), Guard Stop (Stopped (Speed 100%)), and a close button. Below the toolbar, the title bar reads "T\_ROB1 in T\_ROB1/Module1/main". The main window has three tabs: "Tasks and Programs", "Modules" (which is highlighted with a red circle), and "Routines". The code editor displays the following program:

```
69 PROC main ()  
70     !home;  
71     !CutCubeInsideCube;  
72     Routine1;  
73 ENDPROC  
  
74 !     MoveL R3_PrePickPosition2,v100,fine,  
75 !     MoveL R3_PrePickPosition,v100,fine,G  
76 !     MoveTo_R2_PrePickPosition;  
77 !     MoveL R3_PrePickPosition,v100,fine,G  
78 !     MoveL R3_PrePickPosition2,v100,fine,  
79 !     MoveL R3_PrePickPosition2,v100,fine,
```

At the bottom of the editor, there are buttons for "Add Instruction", "Edit", "Debug", "Modify Position", and "Hide Declarations". On the right side, there is a status bar showing "ROB\_1" and two small icons. The bottom left corner of the window shows the path "T\_ROB1 Module1".

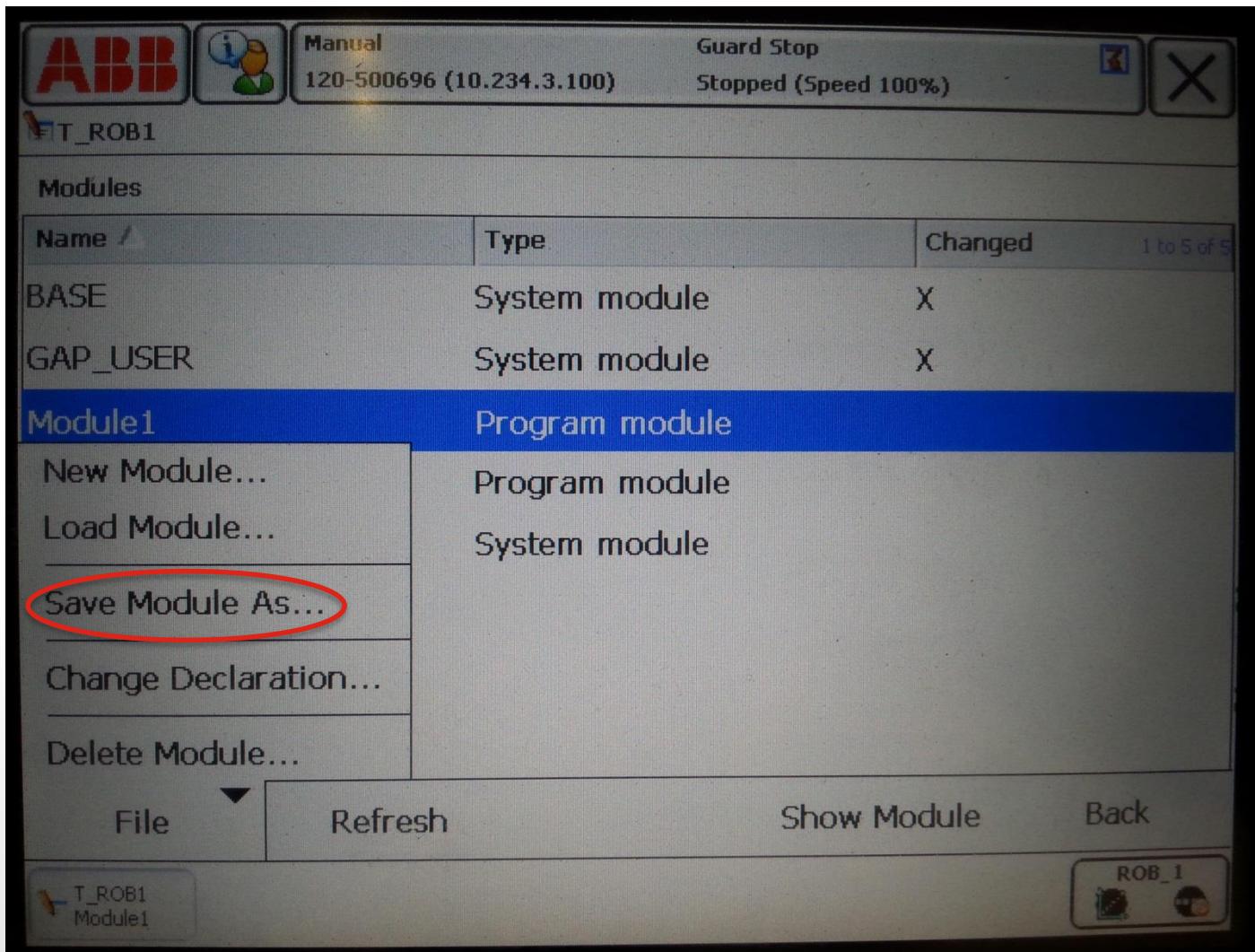
# Backup and Delete Old Program

- A list of modules will appear, including Module1. Press **File**.



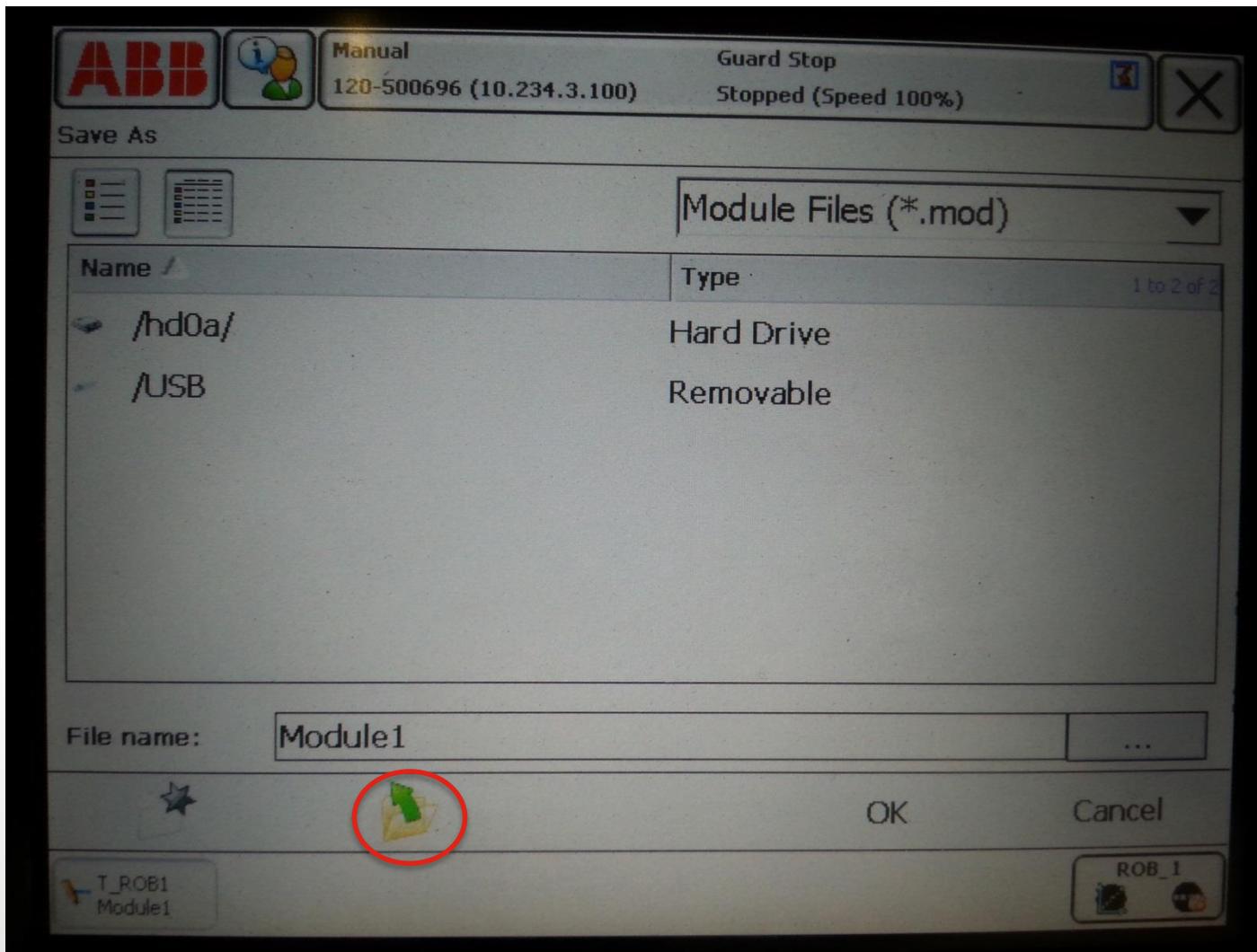
# Backup and Delete Old Program

- From the pop-up menu, choose “Save Module As...”



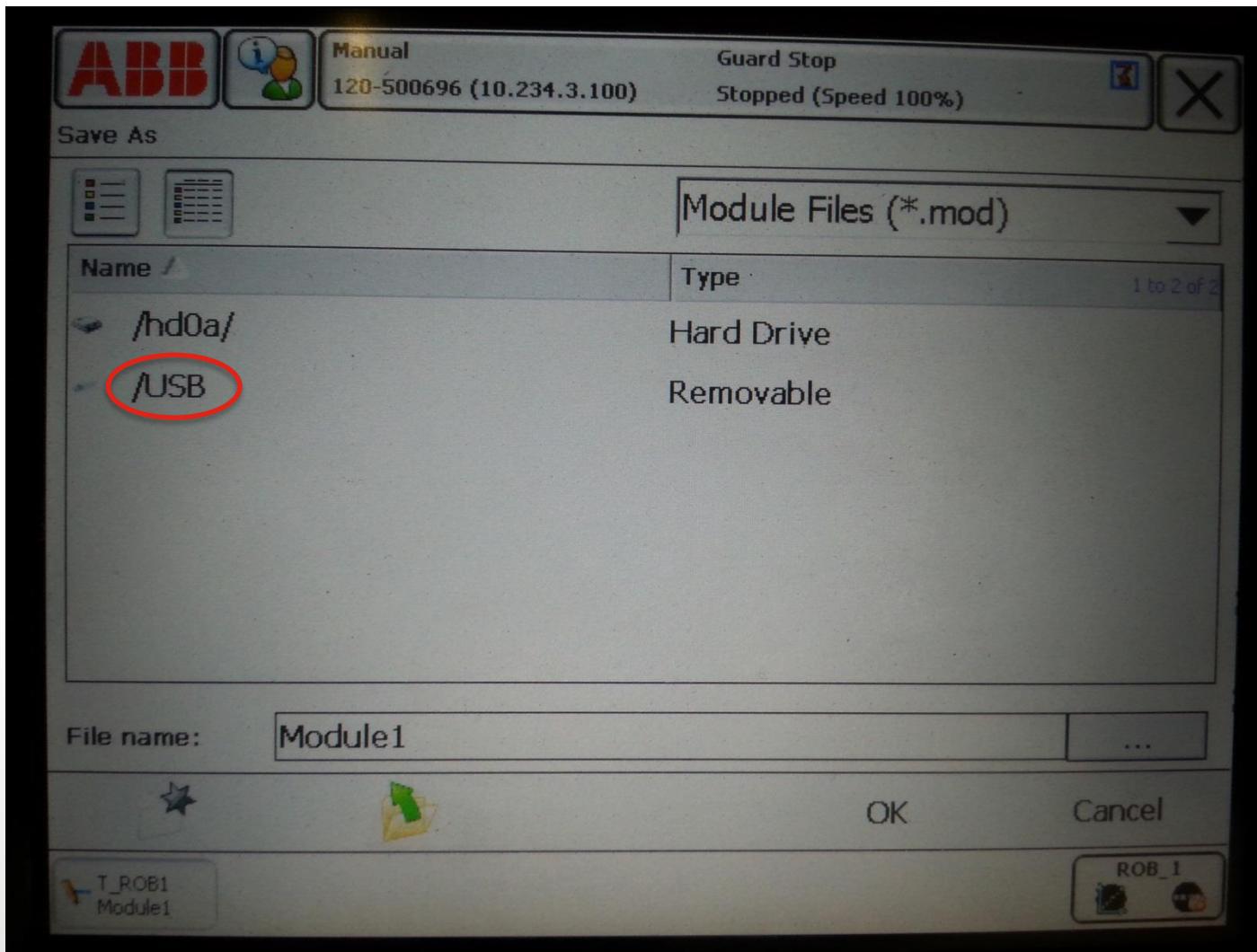
# Backup and Delete Old Program

- Tap the **button** as shown below until we see the page with **/USB**.



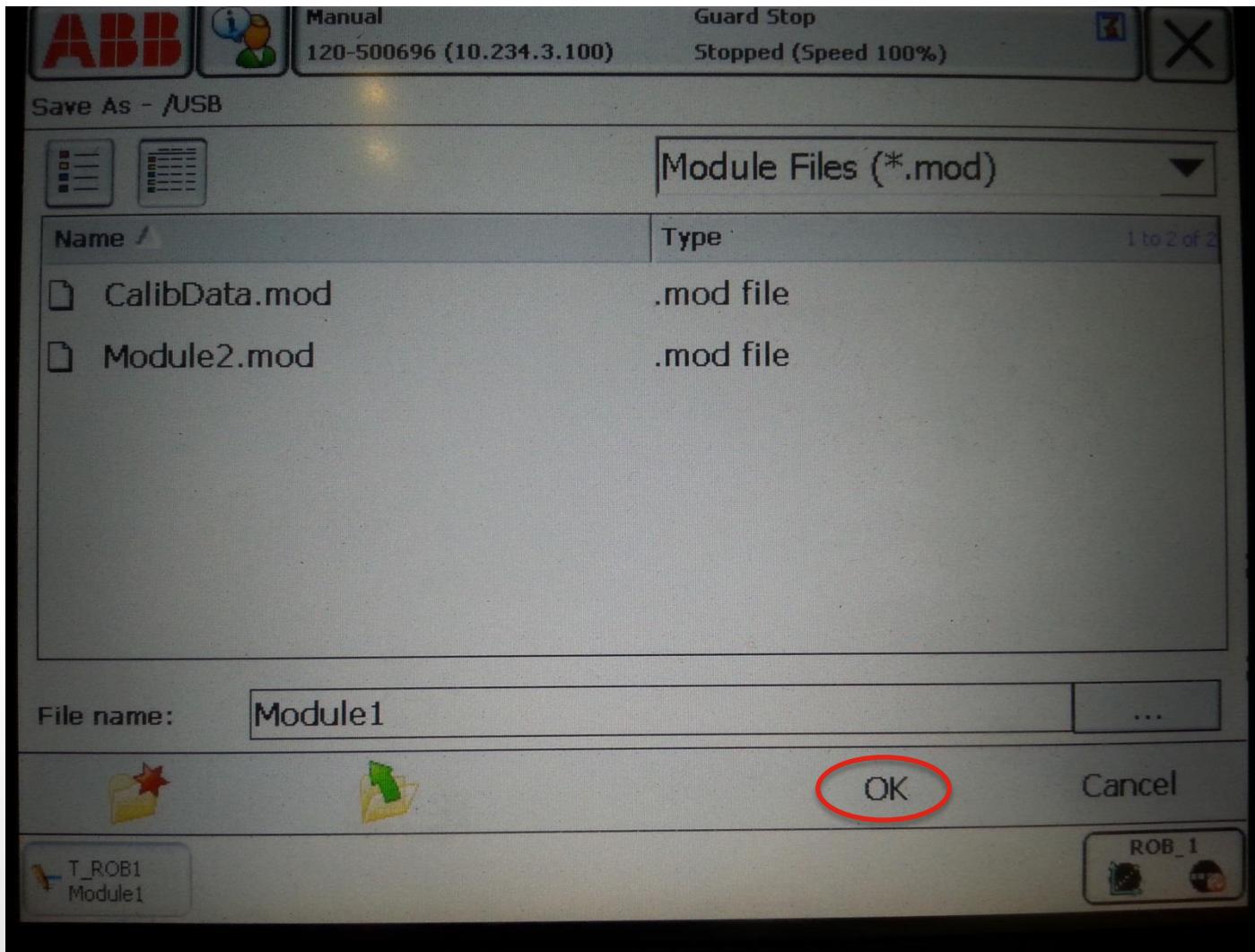
# Backup and Delete Old Program

- Tap on /USB.



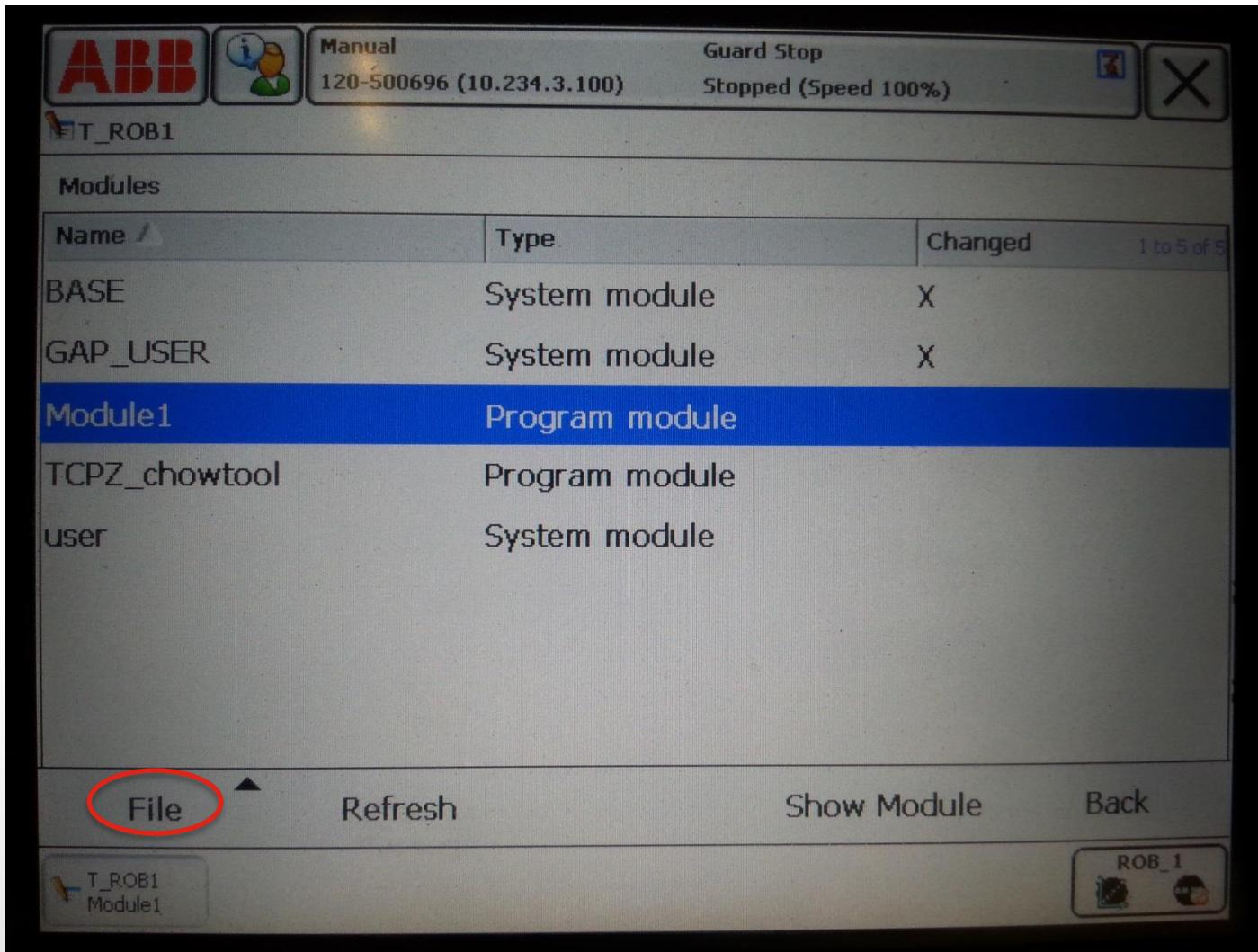
# Backup and Delete Old Program

- Click “OK” and Module1.mod is now copied into the thumb drive.



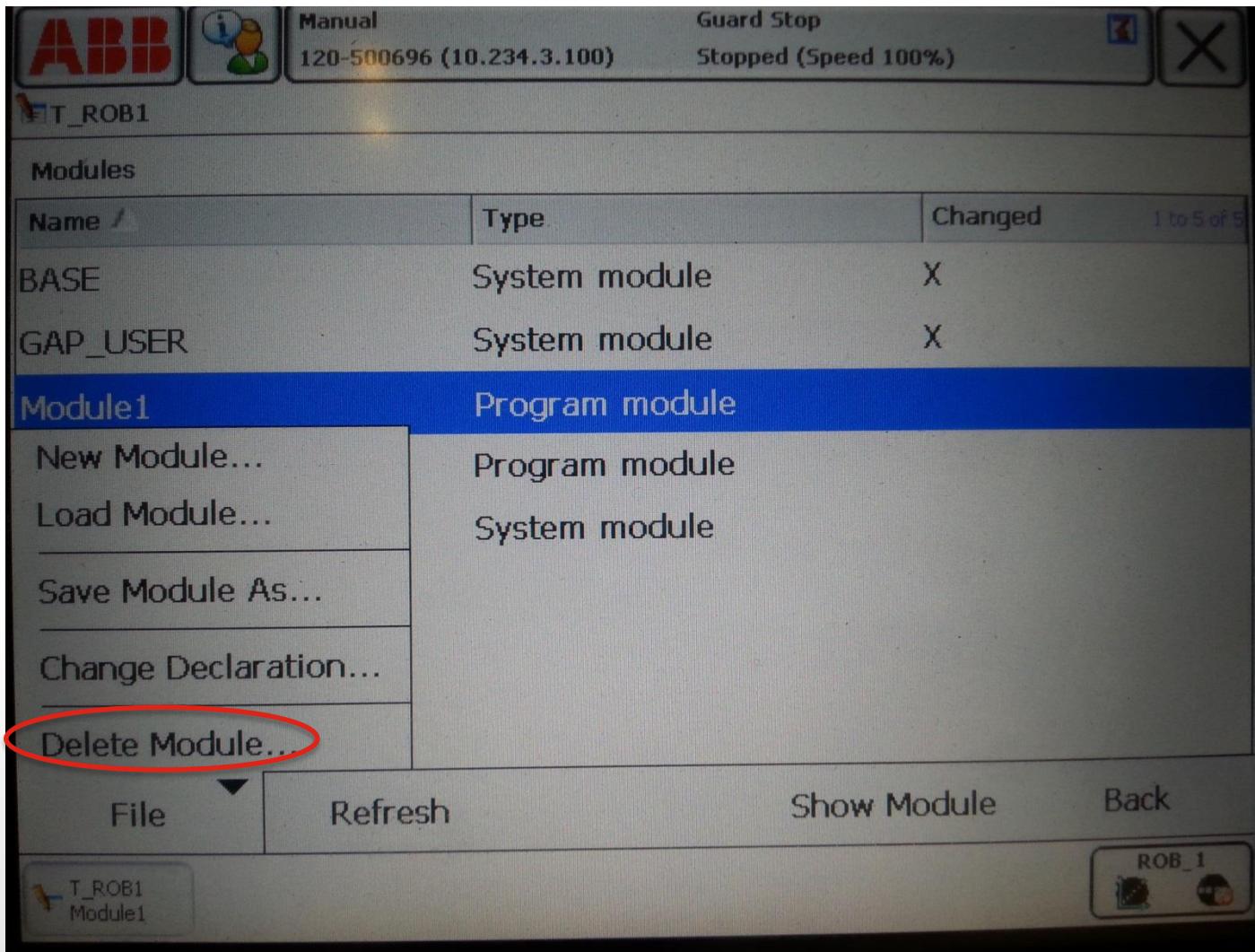
# Backup and Delete Old Program

- Go back to the page with the **list of modules** in robot controller. Click “File”.



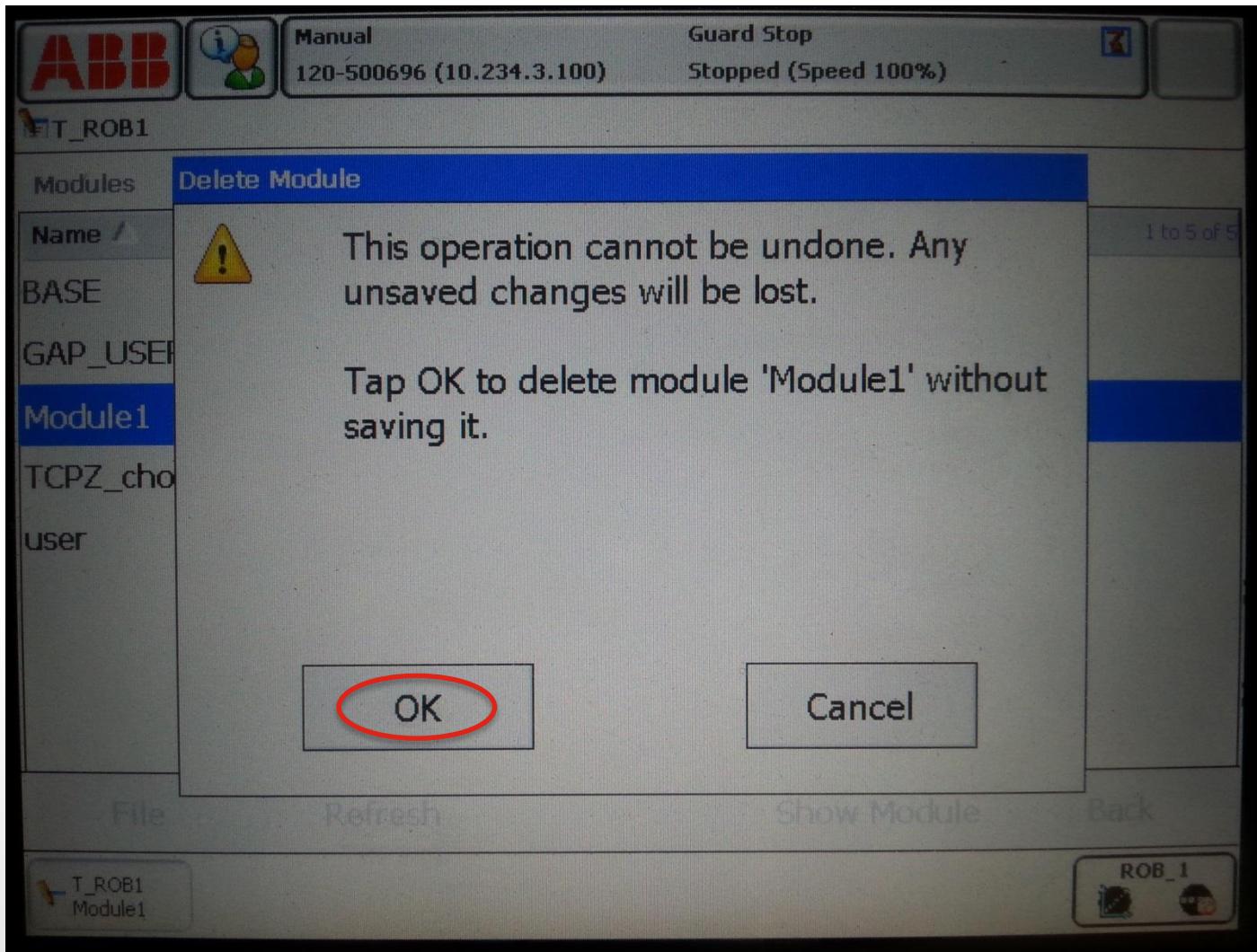
# Backup and Delete Old Program

- From the pop-up menu, choose “Delete Module”.



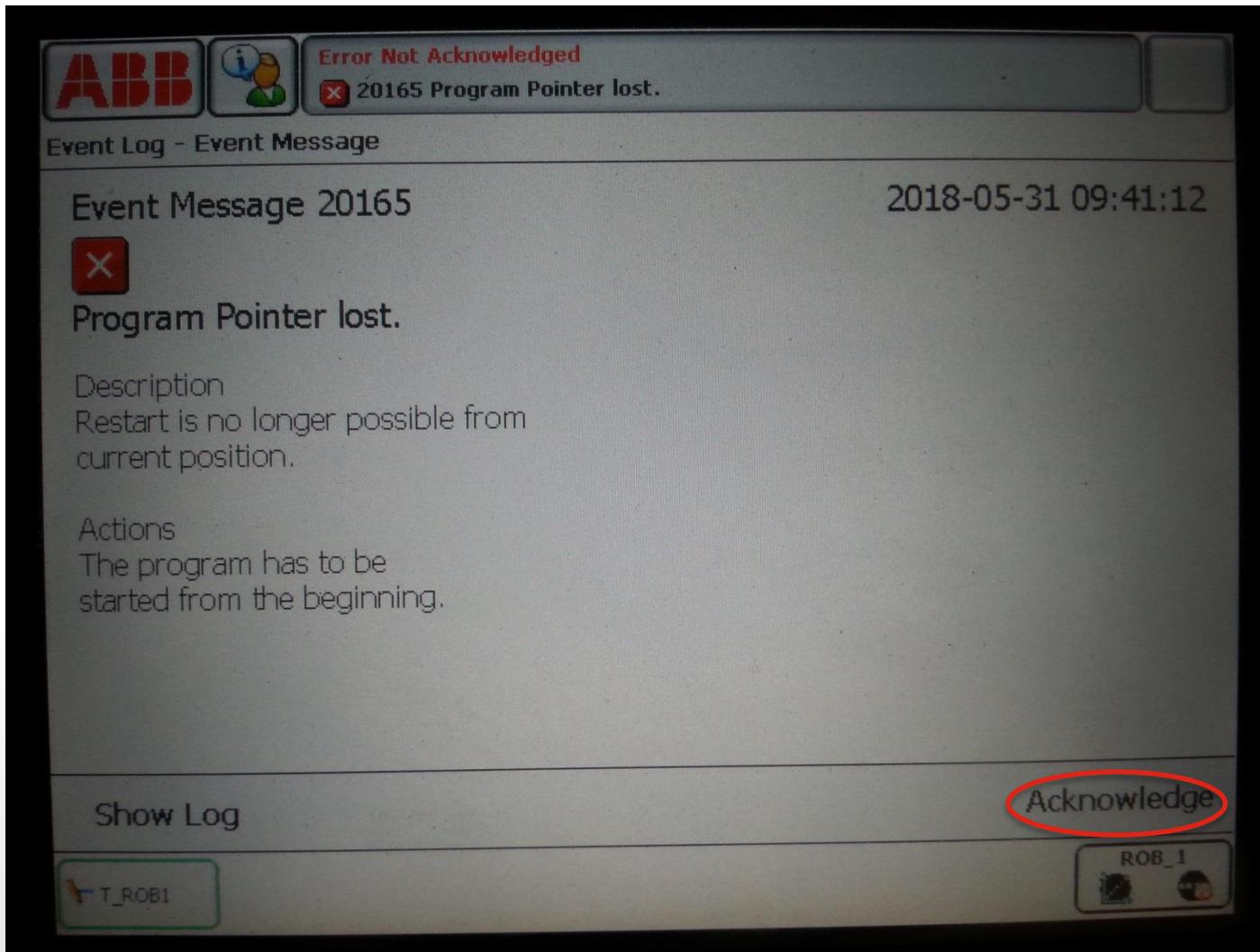
# Backup and Delete Old Program

- There will be a warning message. Click “OK”.



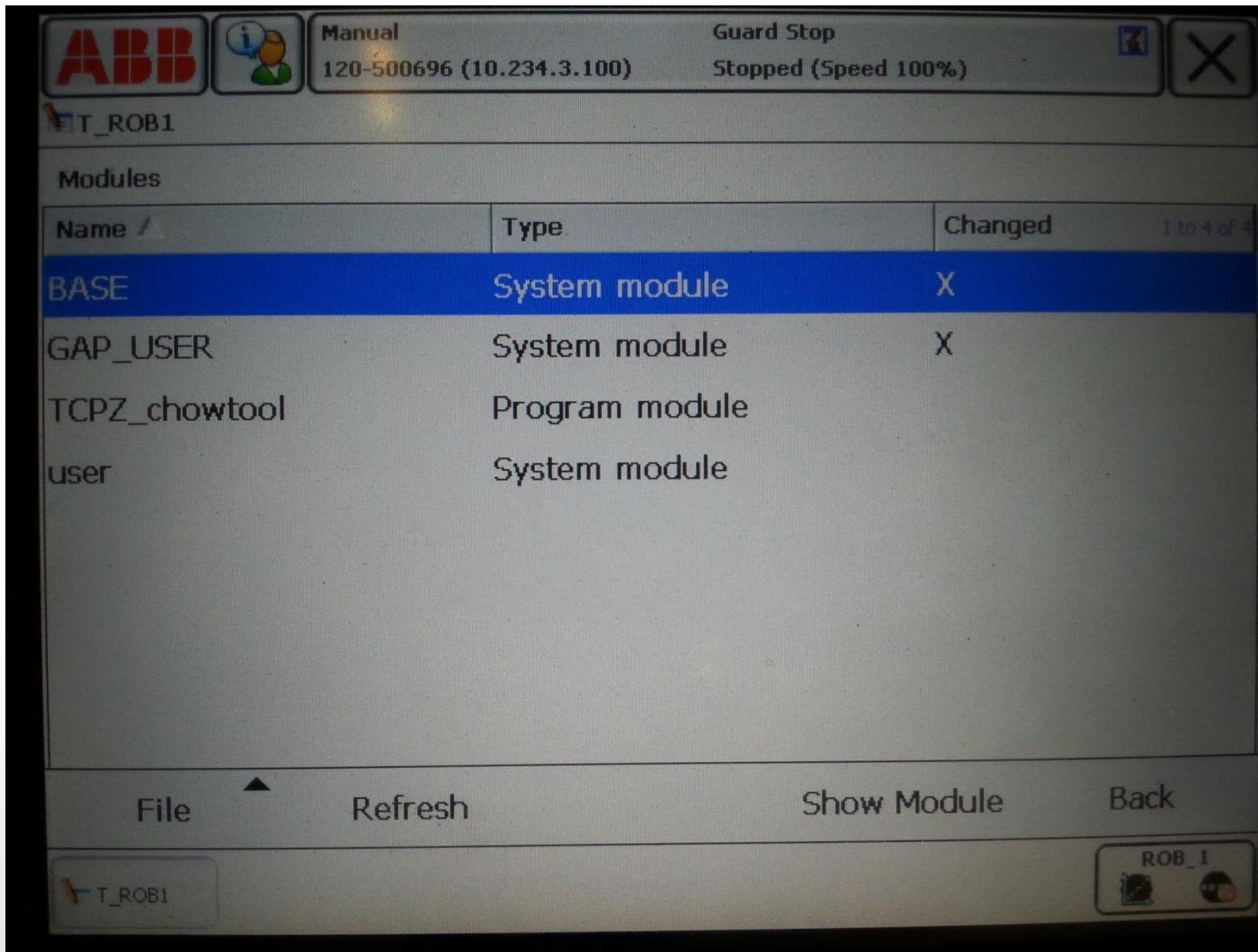
# Backup and Delete Old Program

- The Program Pointer is lost. Click “Acknowledge”.



# Backup and Delete Old Program

- “Module1.mod” is now removed from the list of modules.

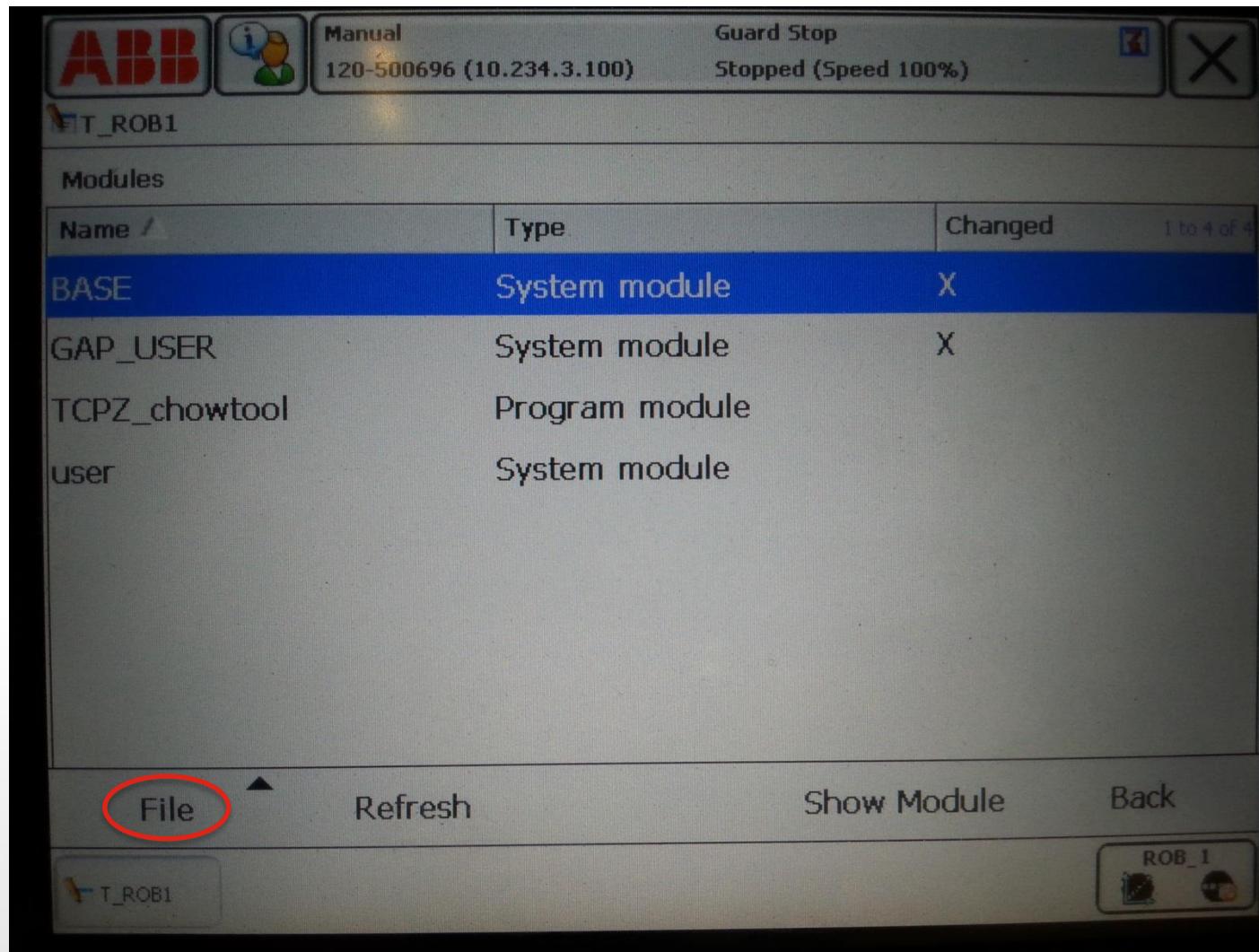


# Content

- Introduction
- Backup and Delete Old Program
- **Load New Program**
- Run New Program
- Delete New Program and Reload Old Program (Optional)

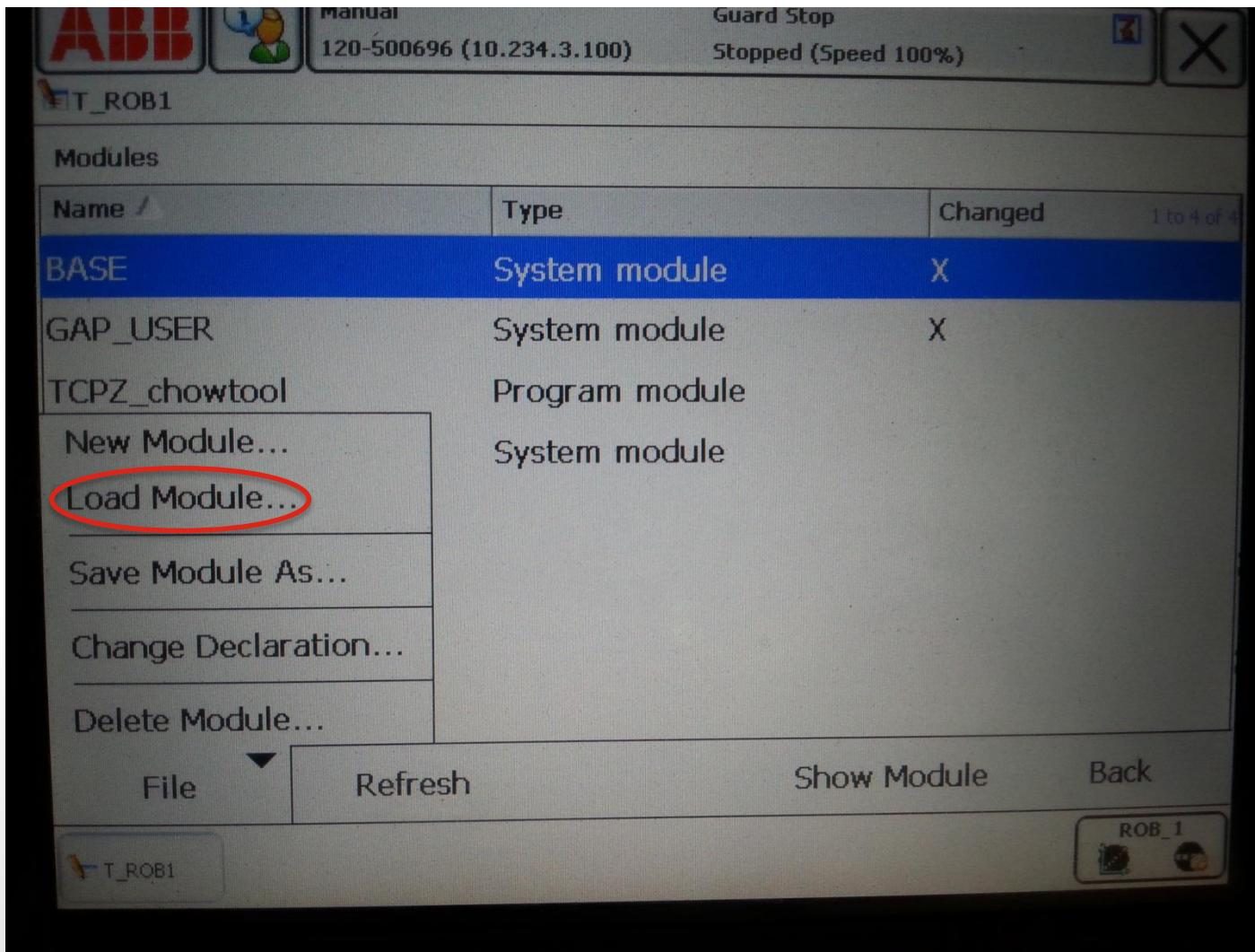
# Loading New Program

- Next, we will load the new program. Click “File”.



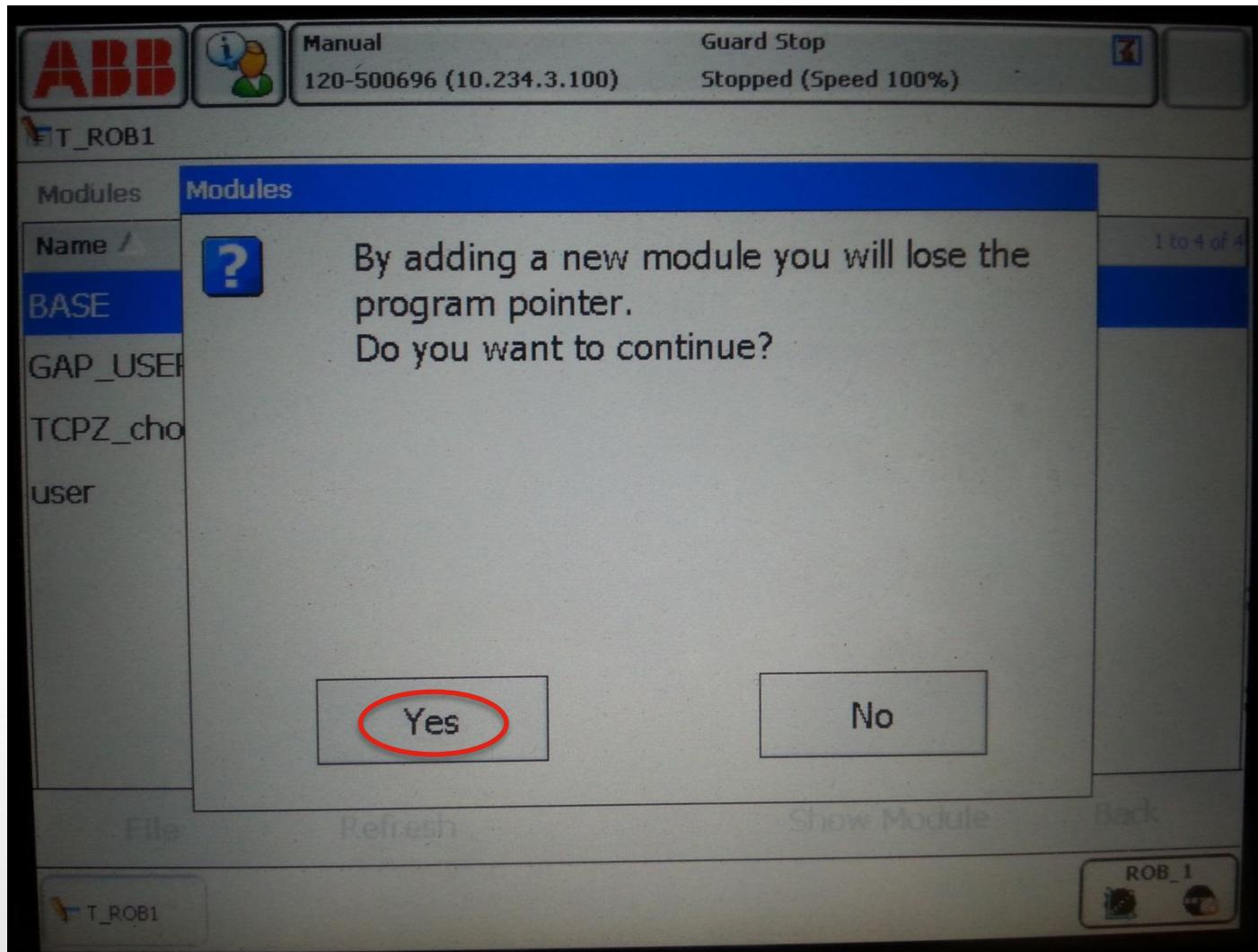
# Loading New Program

- From the pop-up menu, choose “Load Module”.



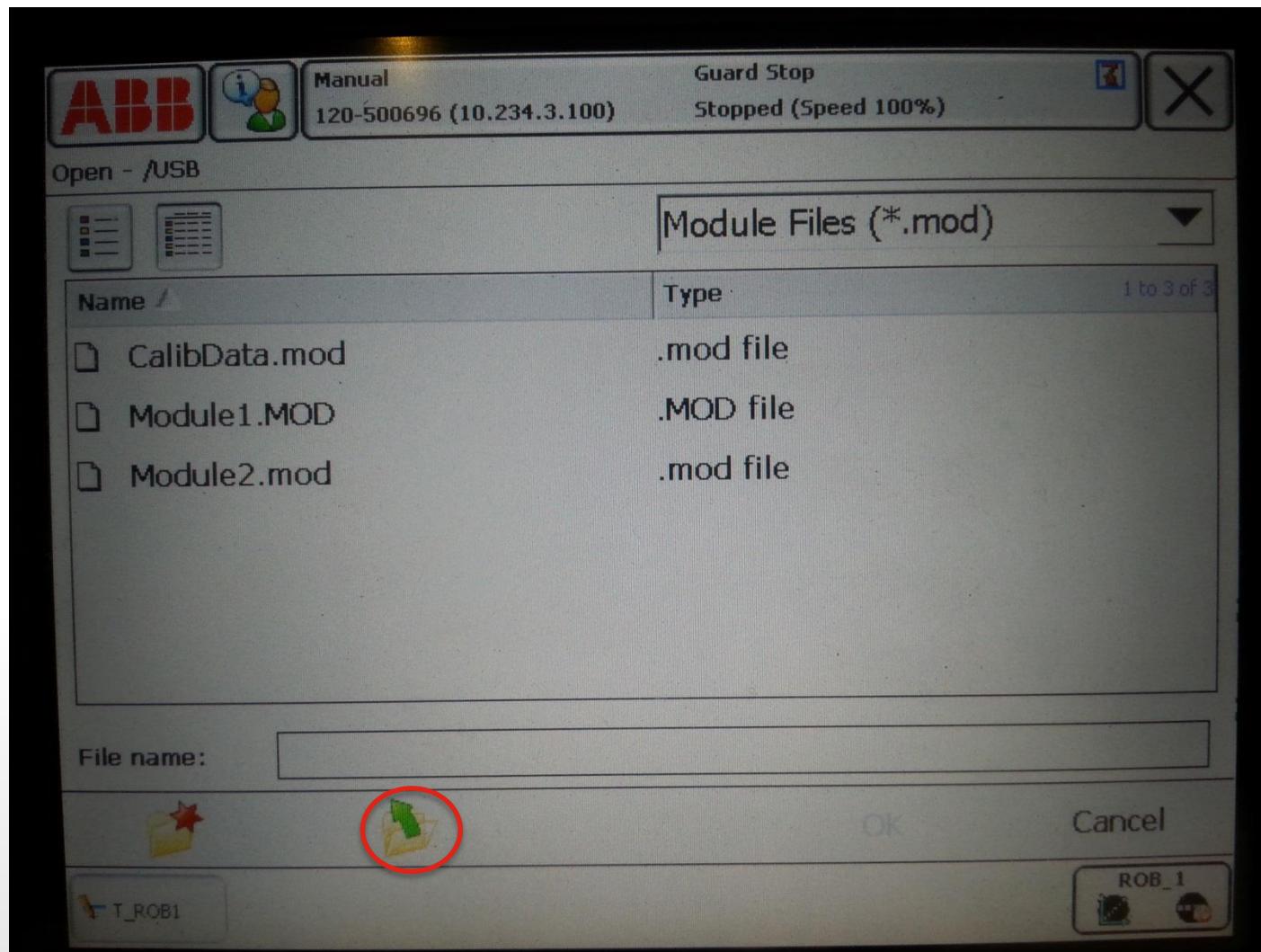
# Loading New Program

- There will be a warning message. Click “Yes”.



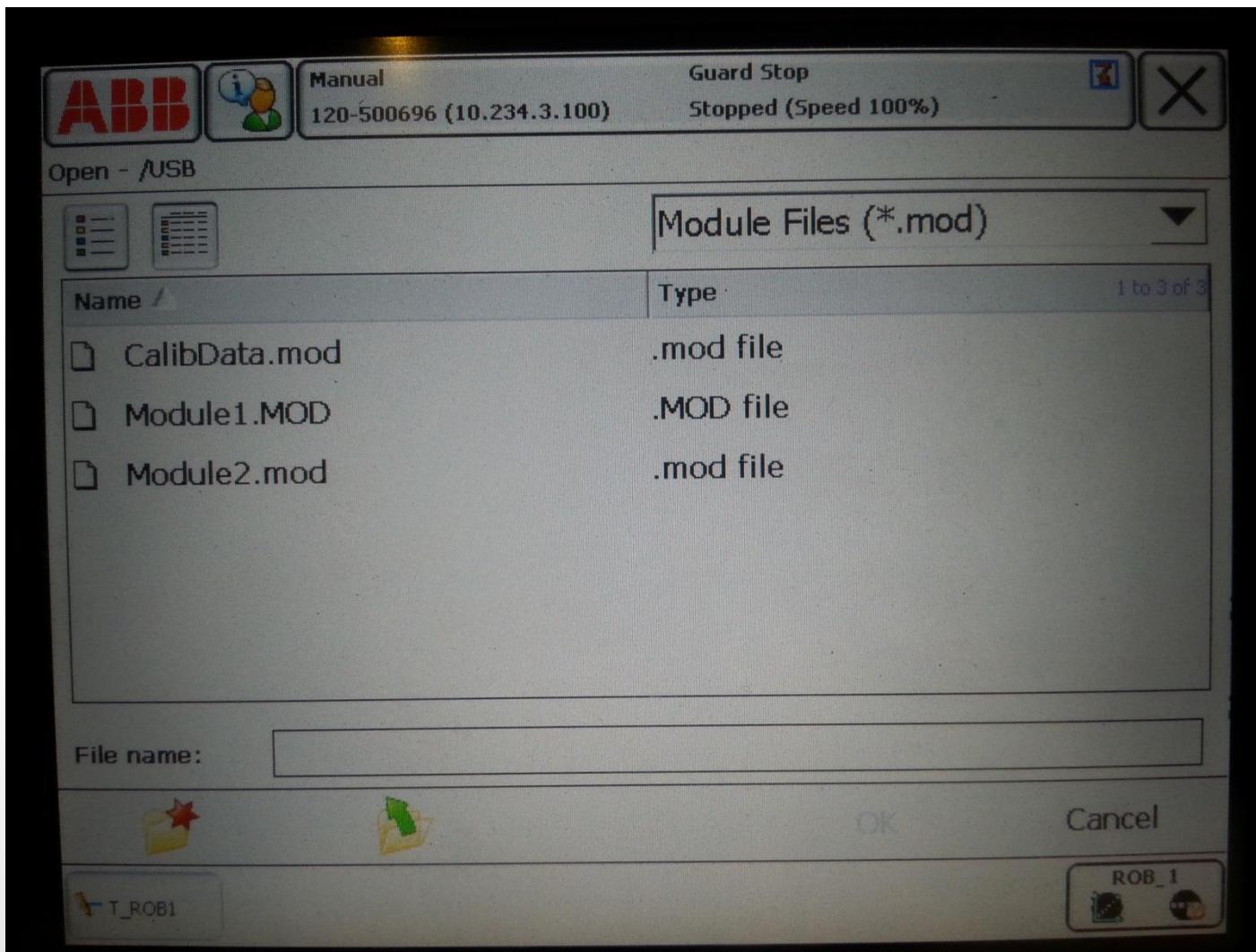
# Loading New Program

- Again, use the “open file” symbol to go to **USB drive**.



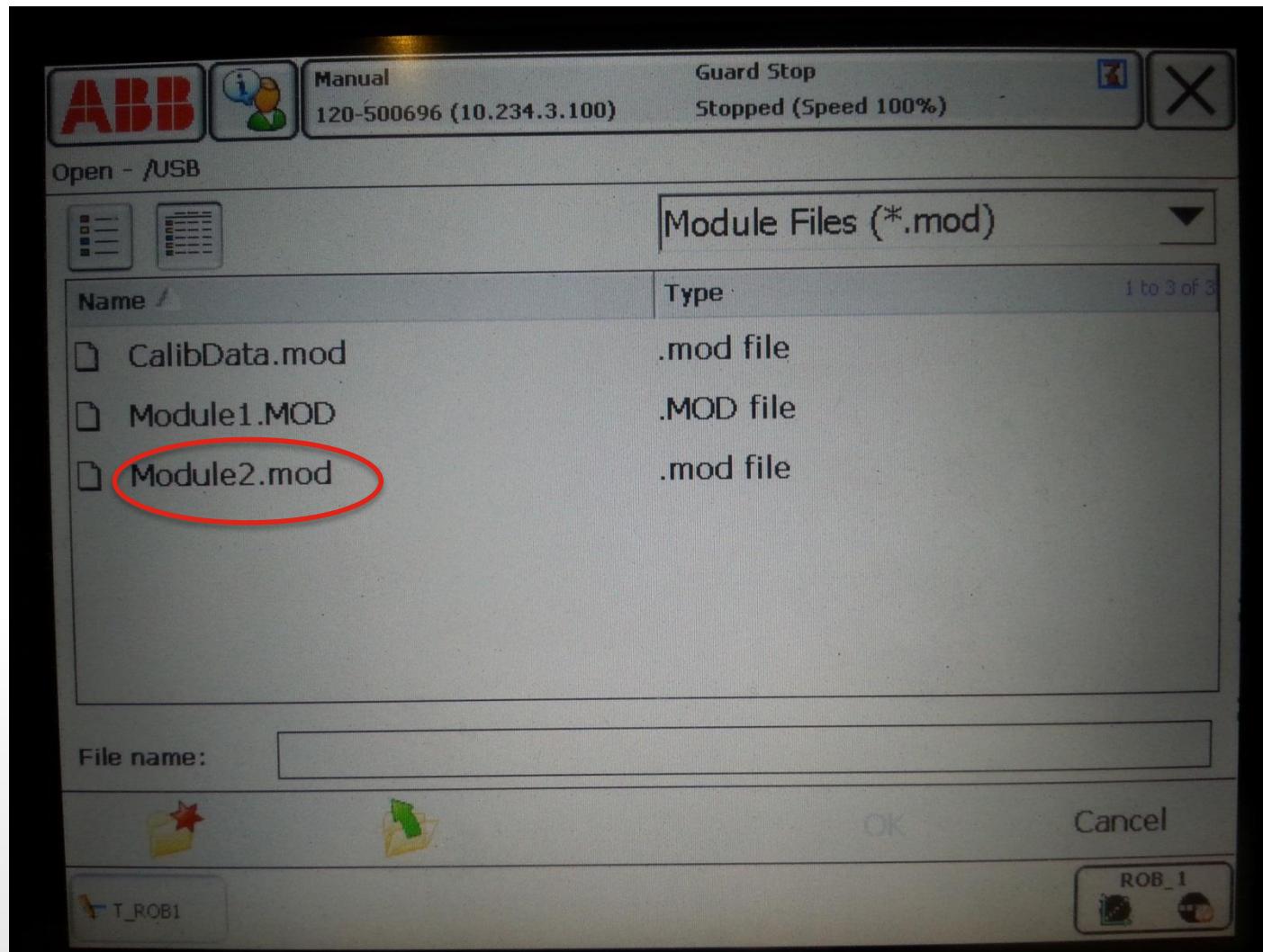
# Loading New Program

- Note: We do not need to load CalibData.mod because the robot controller already has the tool and work-piece data.



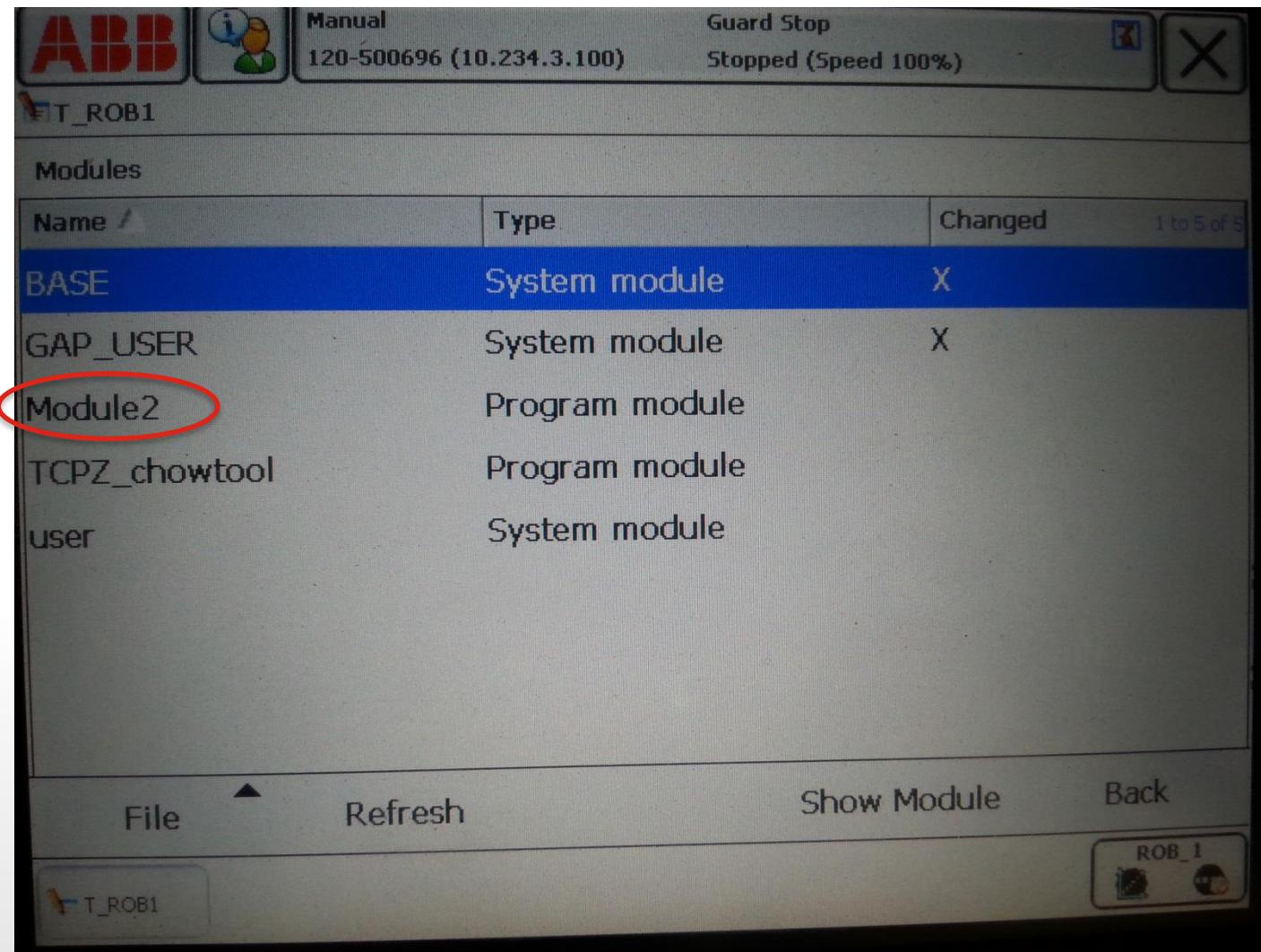
# Loading New Program

- Select “Module2”.



# Loading New Program

- Module2 is now in the list of modules.



# Content

- Introduction
- Backup and Delete Old Program
- Load New Program
- Run New Program
- Delete New Program and Reload Old Program (Optional)

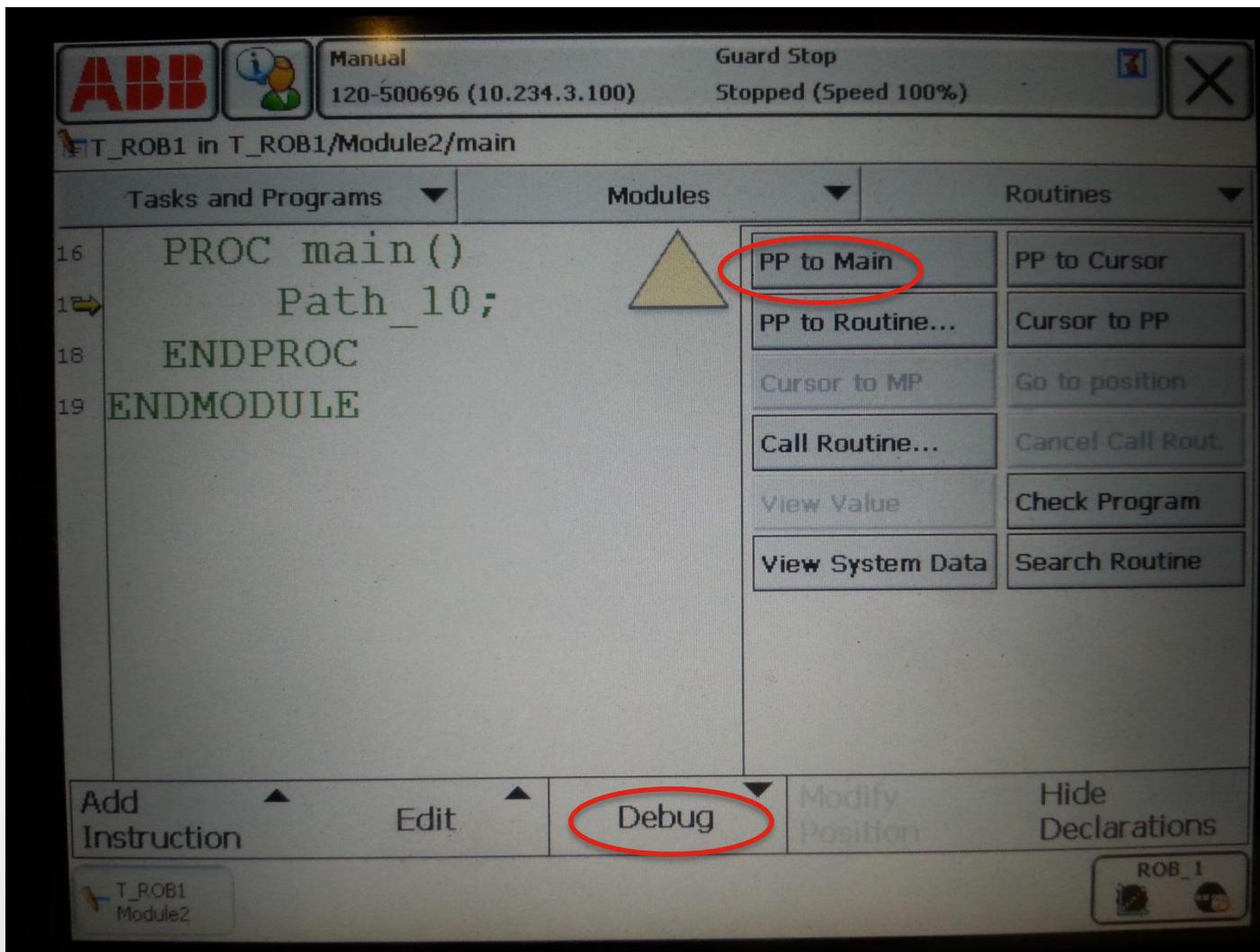
# Run New Program

- Go back to **Program Editor** and we will see that **Module2** is active.

```
16 PROC main()
17     Path 10;
18 ENDPROC
19 ENDMODULE
```

# Run New Program

- Click “Debug” then “PP to Main”.



# Run New Program

- Hold the “motor on” button and press “play” on pendant. The robot will run according to the program.



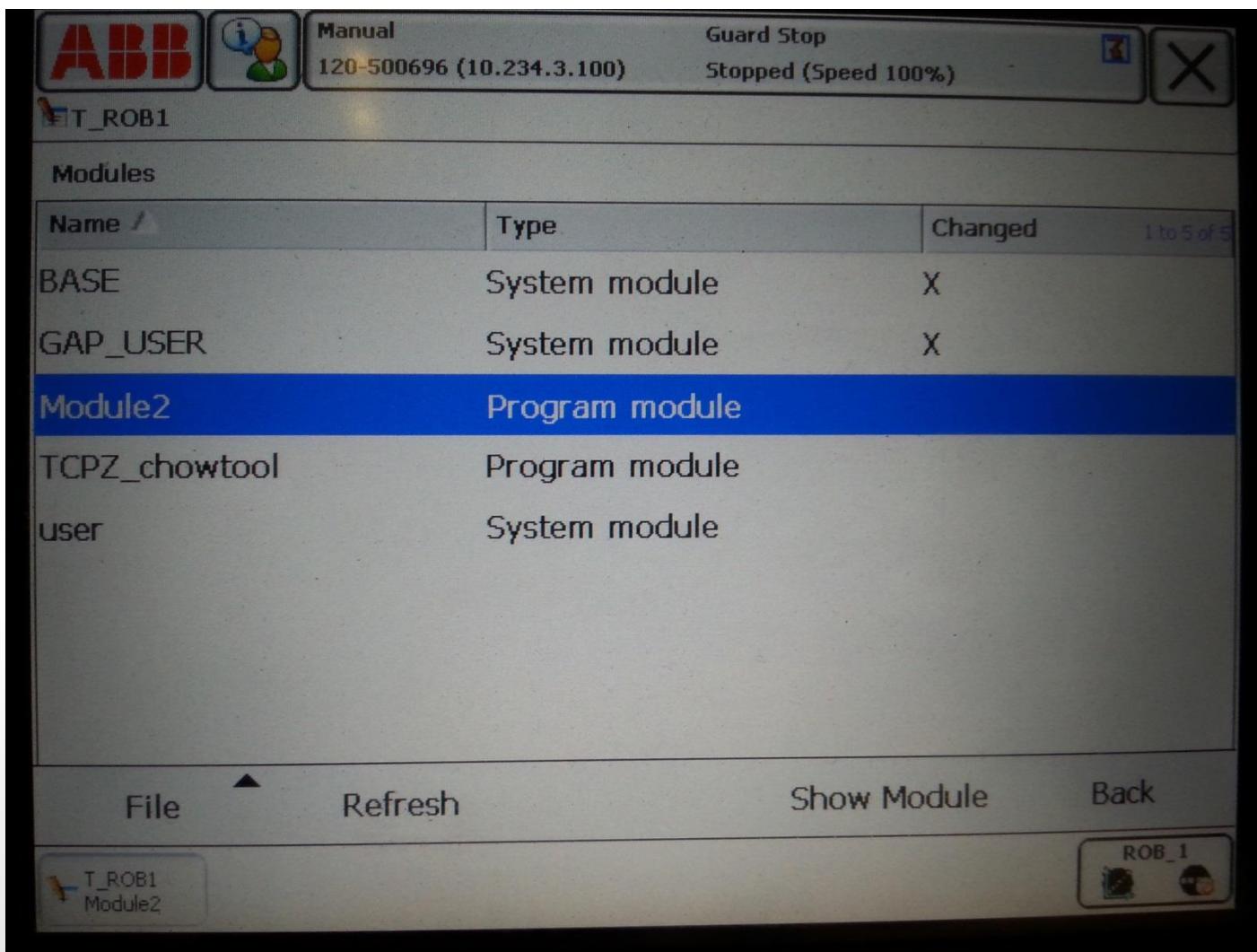
# Content

- Introduction
- Backup and Delete Old Program
- Load New Program
- Run New Program
- Delete New Program and Reload Old Program (Optional)

# Delete New & Reload Old Program

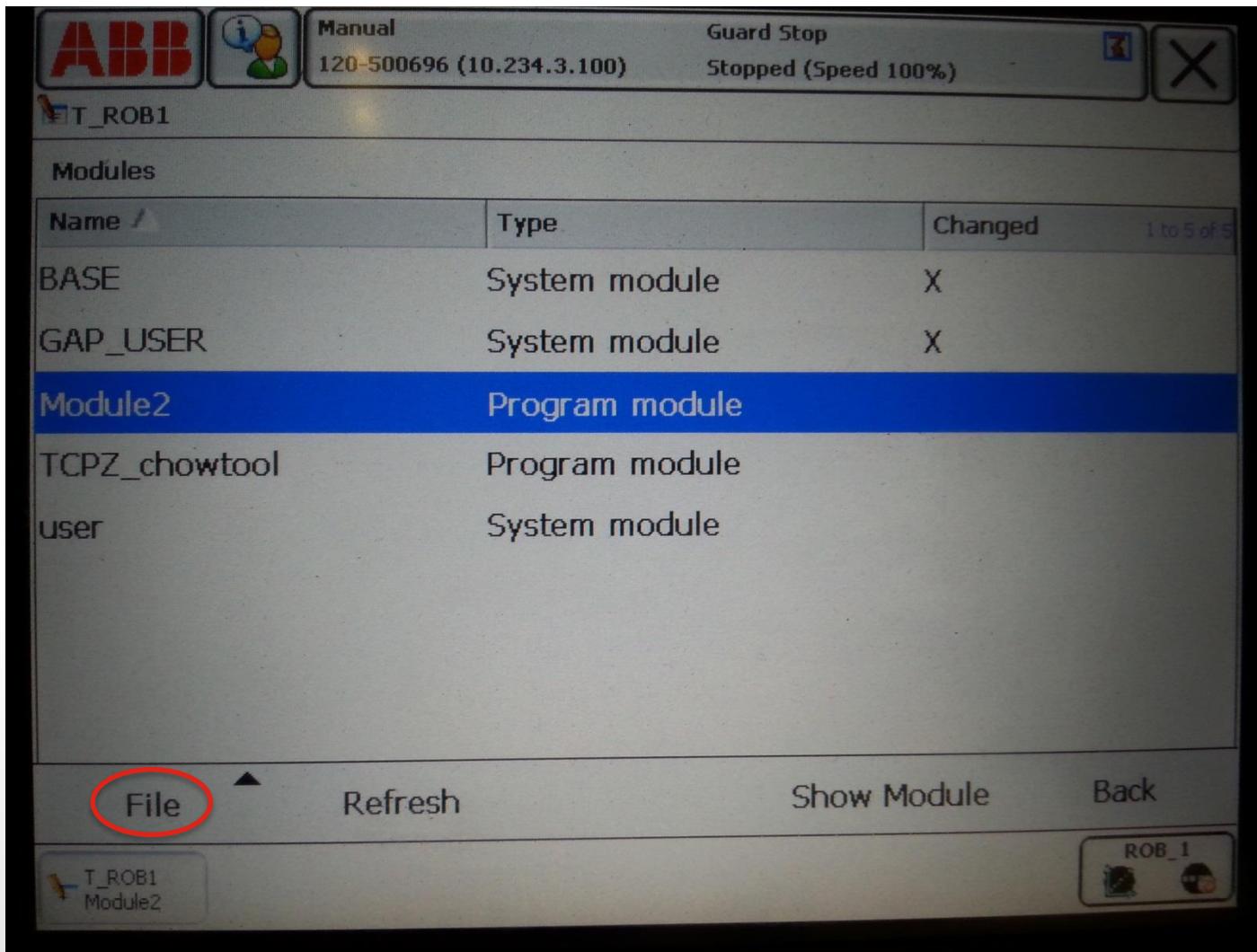
- We can switch back to the original program. In **program editor**, click “**Module**”

and the list of modules is shown.



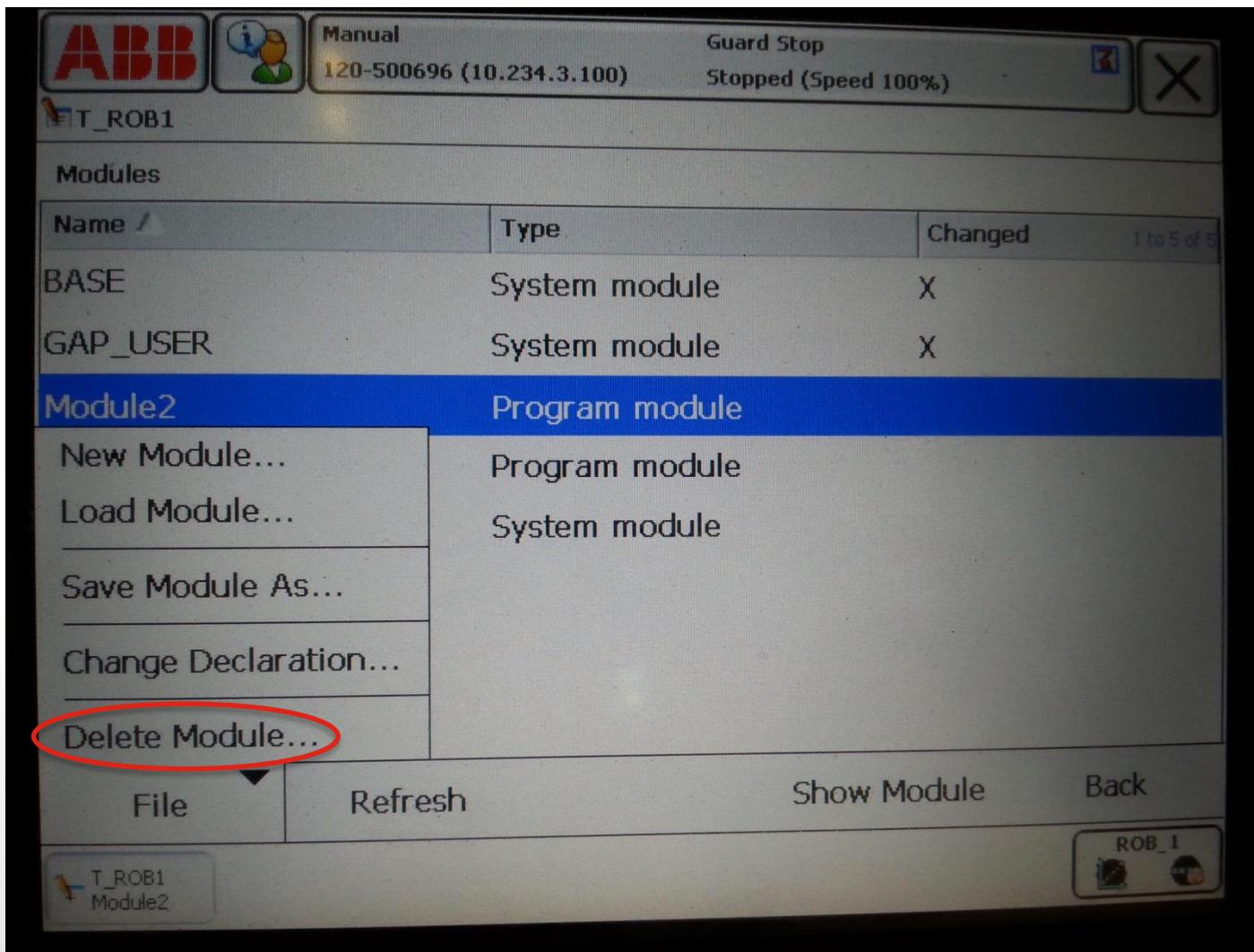
# Delete New & Reload Old Program

- Choose Module2 and click “File”.



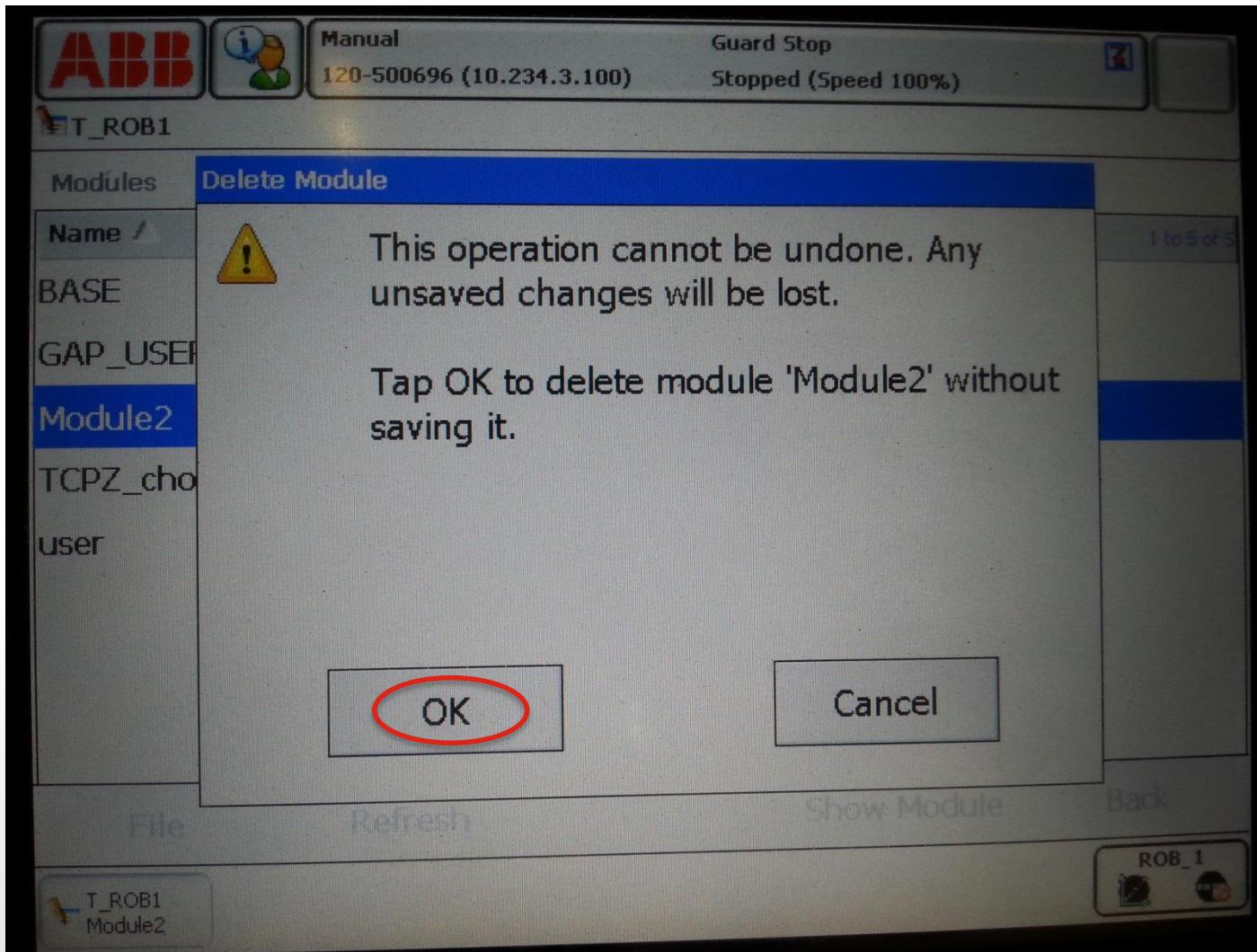
# Delete New & Reload Old Program

- On the pop-up menu, choose “Delete Module”.



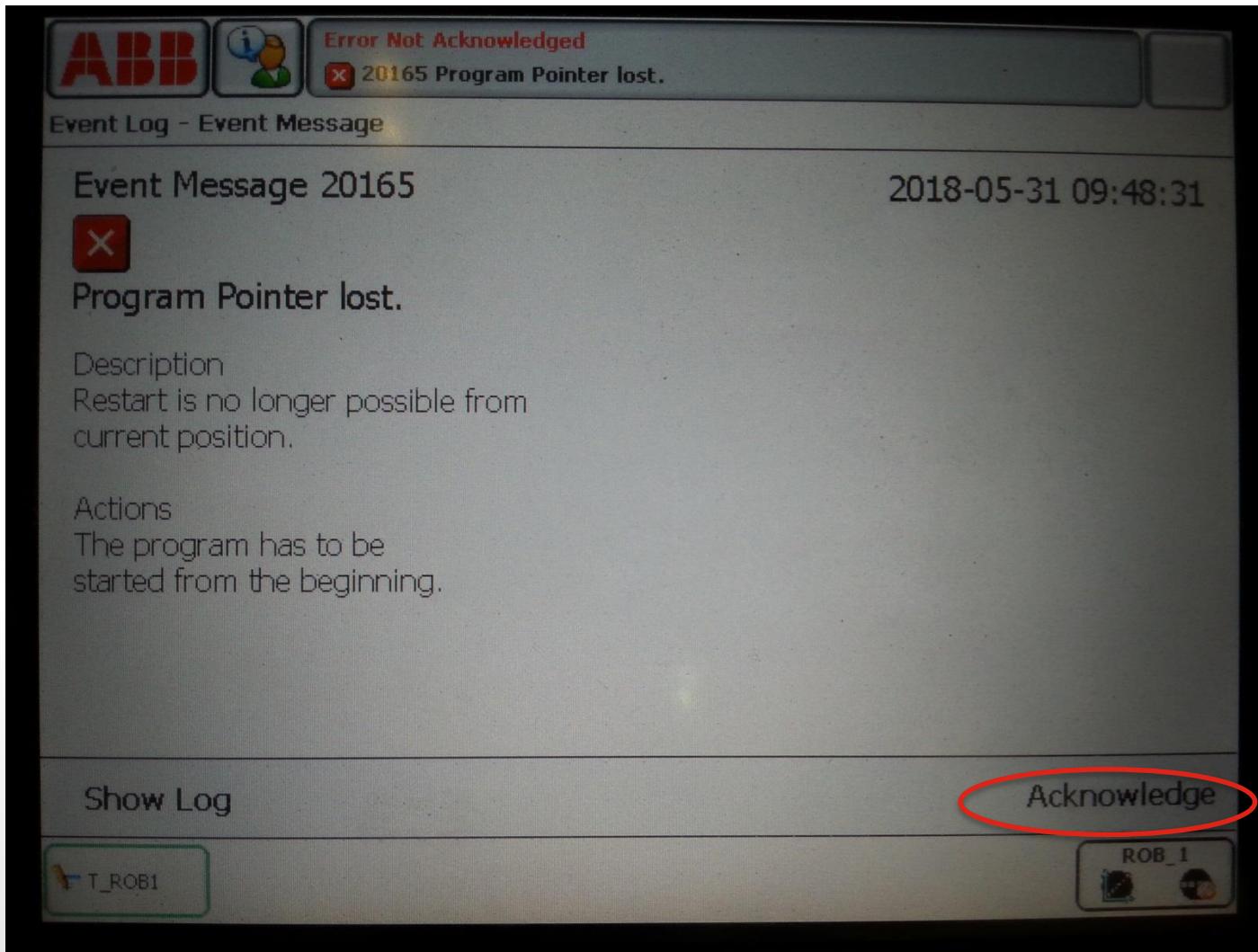
# Delete New & Reload Old Program

- There will be a warning message. Click “OK”



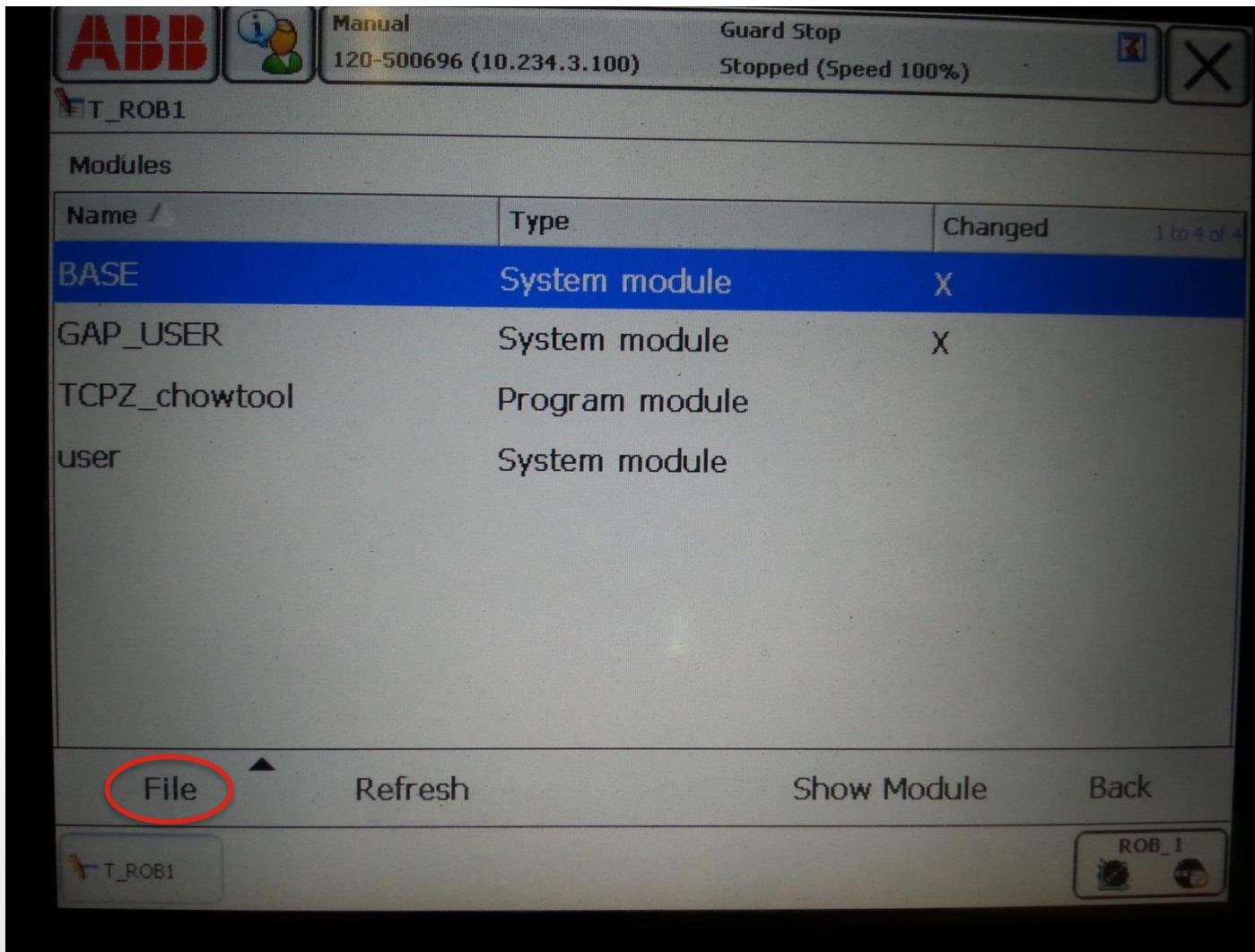
# Delete New & Reload Old Program

- The program pointer is lost. Click “Acknowledge”



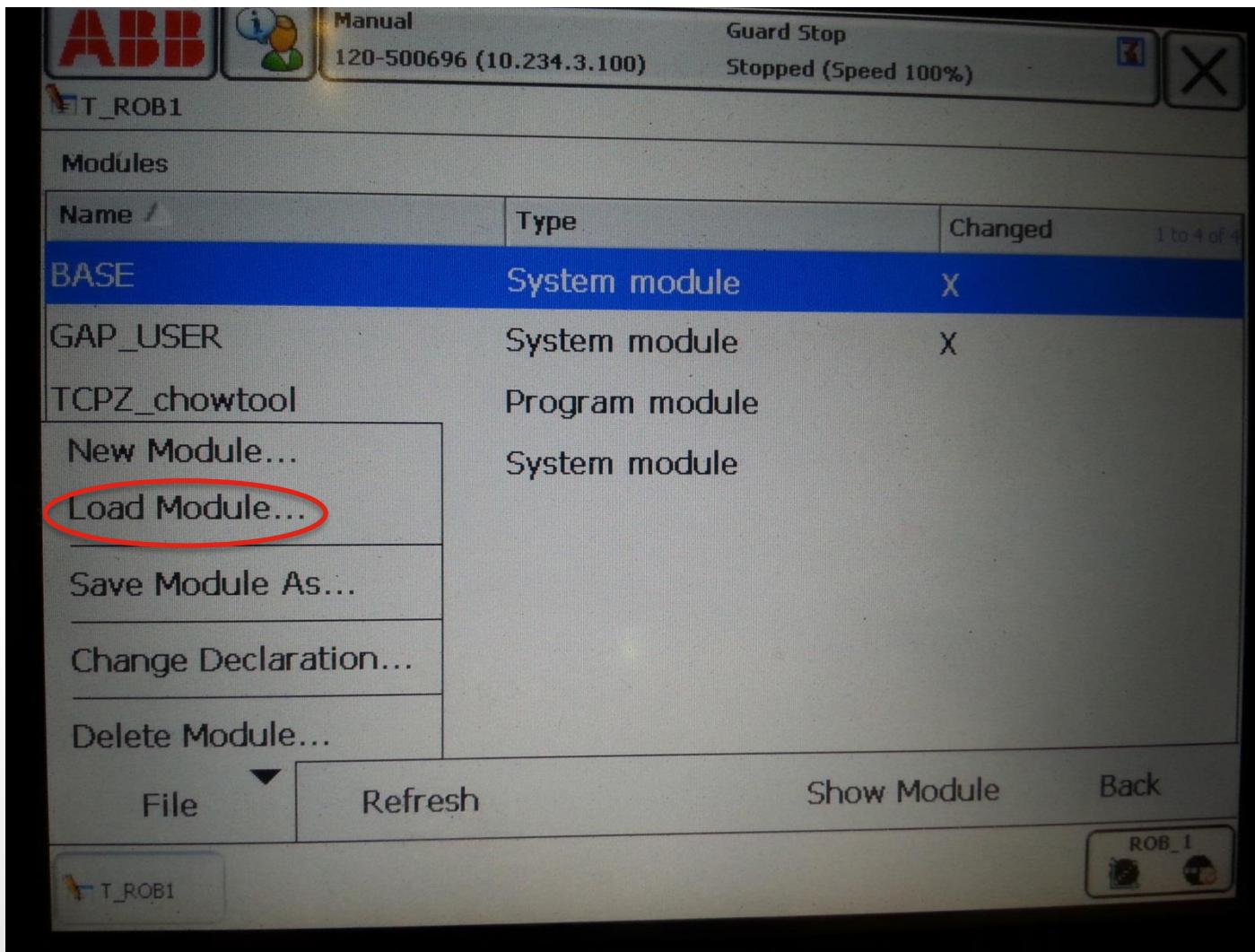
# Delete New & Reload Old Program

- Back at the modules list, click “File”



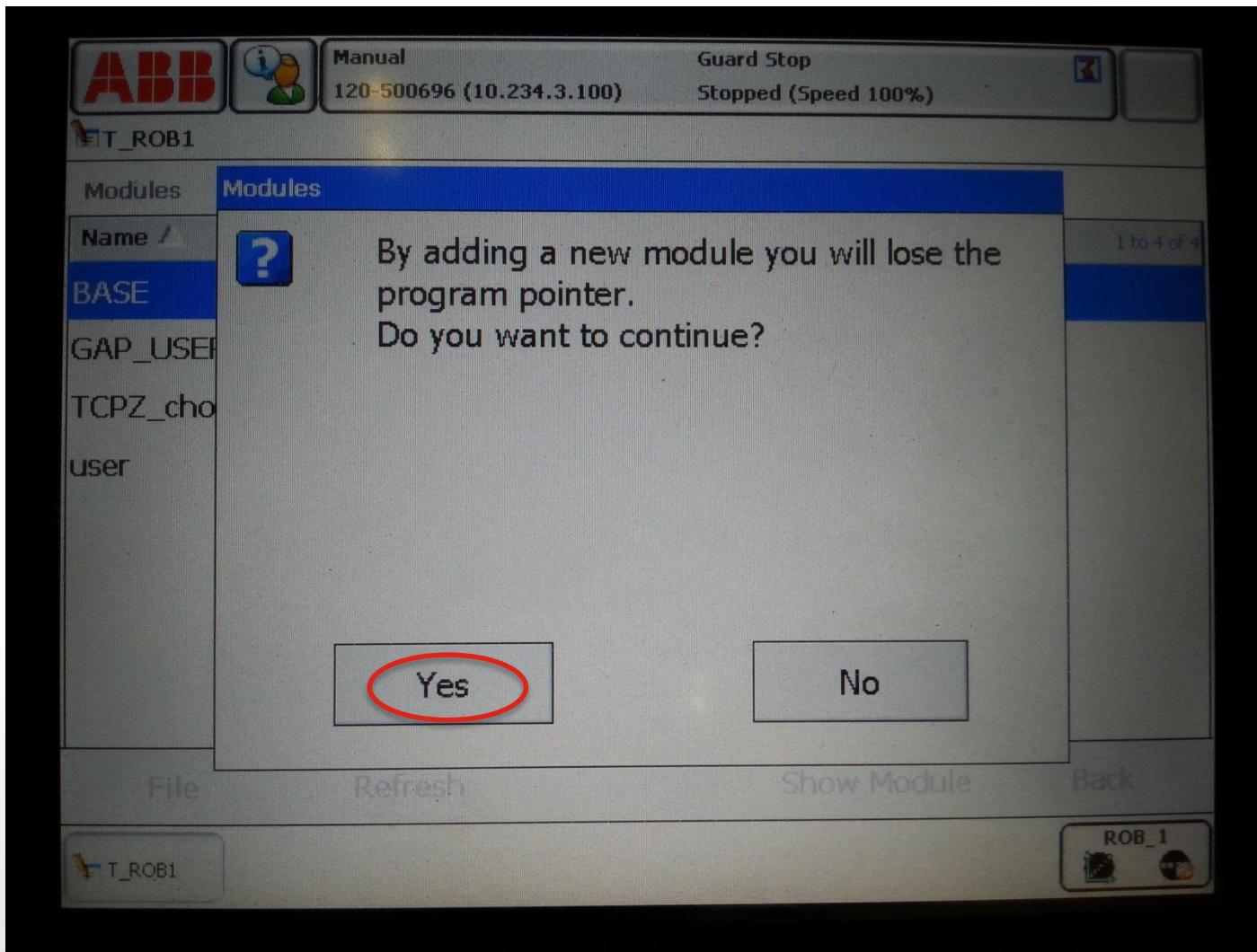
# Delete New & Reload Old Program

- On the pop-up menu, click “Load Module”



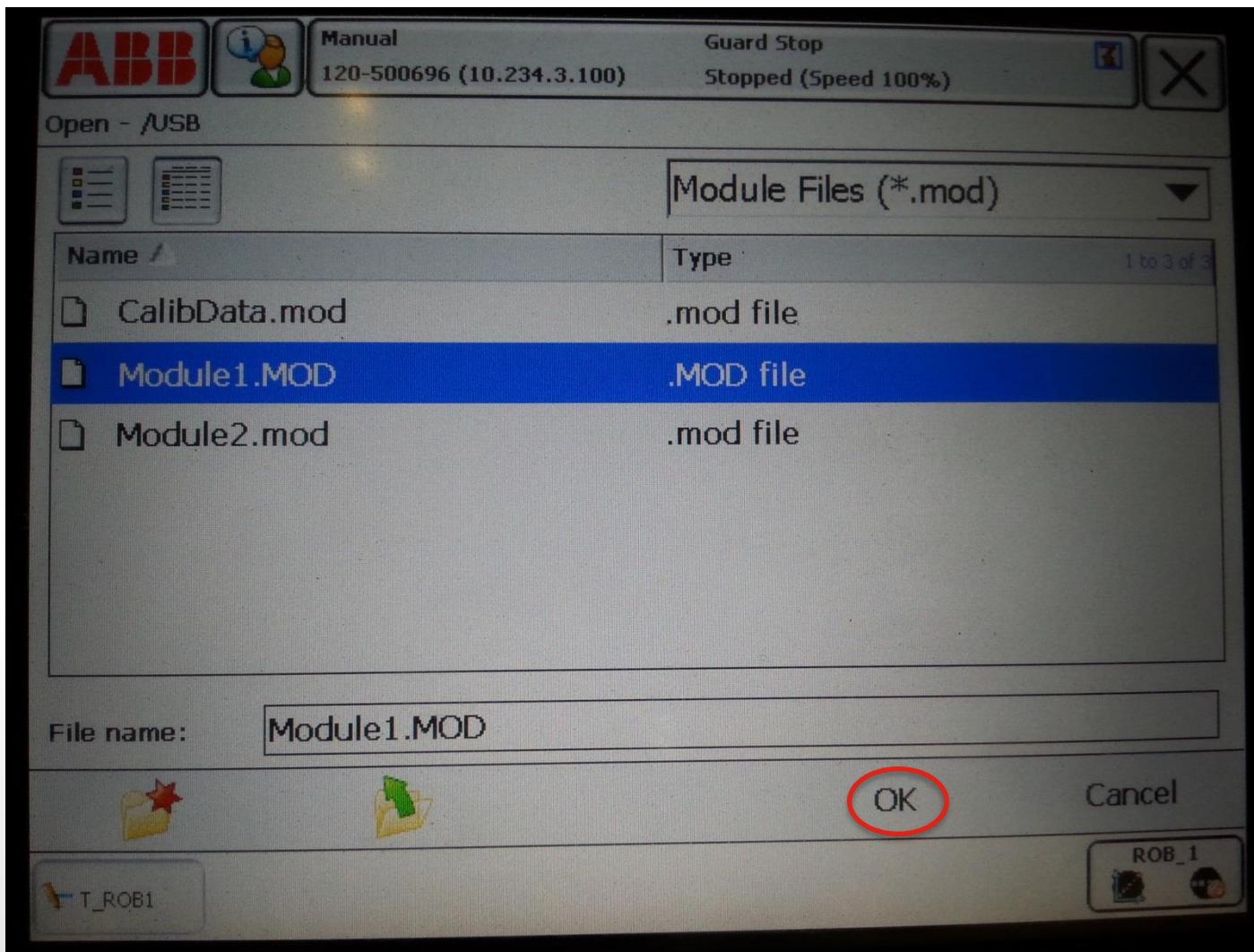
# Delete New & Reload Old Program

- To the warning message, click “Yes”



# Delete New & Reload Old Program

- Highlight Module1 (in USB Drive), and click “OK”. The old program is now loaded.



# Thank you!

---

Have a good evening.

