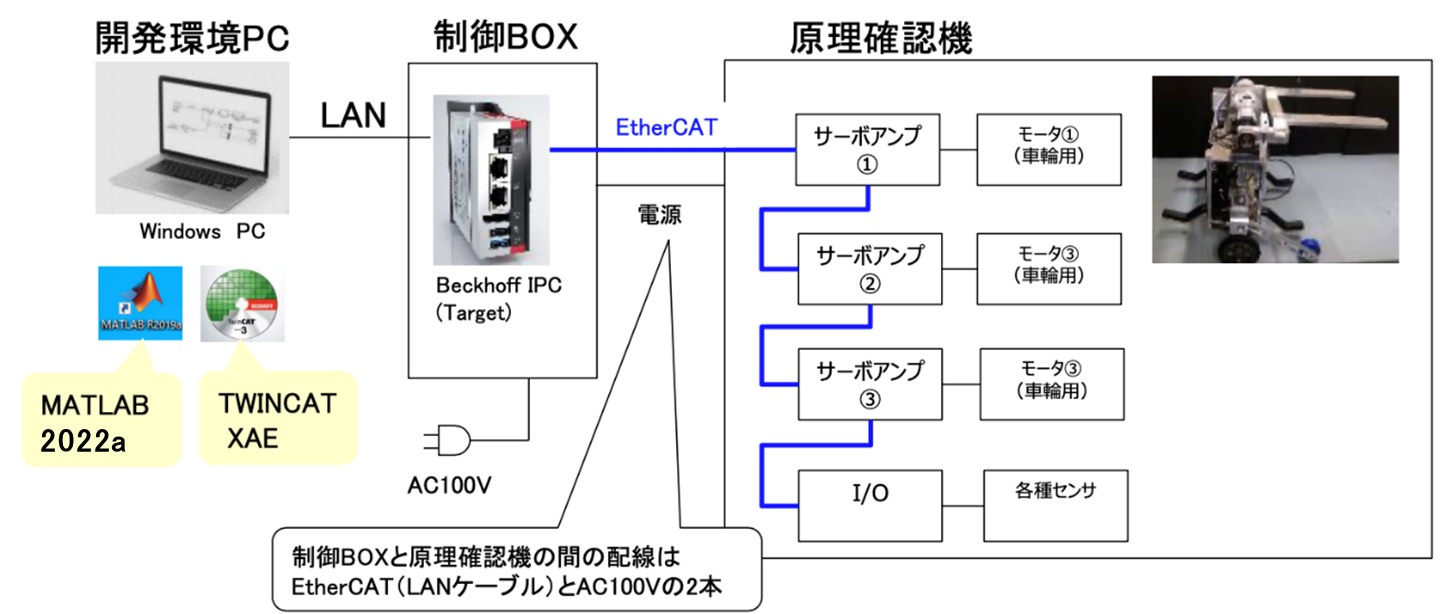
Two-wheeled forklift experiment manual

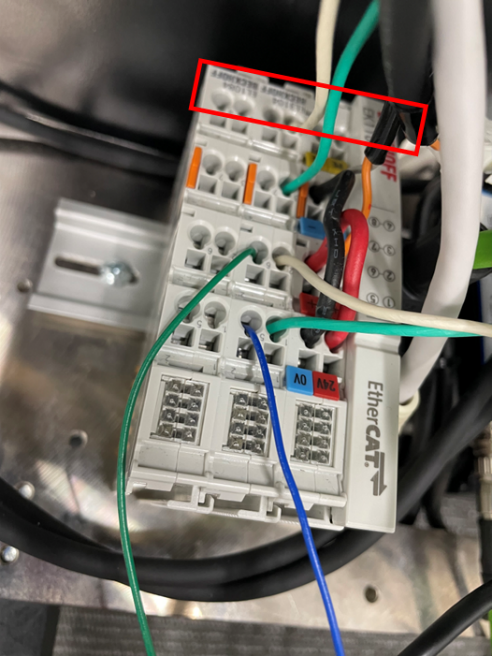
Created by: Hirotake Kanazawa ( Class of 2024 )

The structure has changed slightly since Yajima's time, so here we will summarize the new operation methods for two-wheeled forklifts.

Hardware configuration



The control PC (ELSA VELUGA 3060 G2-17) and the Beckhoff IPC in the control box are connected via a LAN cable. To find out what the sensors etc. are connected to, check the model numbers in the red box in the image below on the Beckhoff website.



The composition is as follows.

|  |  |  |  |
| --- | --- | --- | --- |
| E- L5101 | E L1084 | E- L3104 | E- K1100 |

load cells are connected to the E- L3104 (4 ch A- D unit). If you want to connect other sensors, you can add them here, or if you run out of channels, you can increase the number of channels by purchasing a new unit and adding it to the left end.

Software Configuration

Matlab File​

|  |
| --- |
| TBRobot / 　├ TBRobot\_main.m is the main m file. Run this. 　├ TBRobot\_exturnal19a\_safe.slx Simulink file. 　├ tcfile\_converter.m Used to collect and save experimental data. 　├ functions/ 　│　├ switch\_machine.m Used to change parameters of loom and Keio. 　│　├ calc\_ref\_matrix.m For generating command values for position control. 　│　└ calc\_ref\_element.m Same as above. 　└params / 　　　├ TBRobot\_safety\_params.m Setting safety limits. 　　　├ TBRobot\_model\_params.m Nominal value settings for the model. 　　　└ TBRobot\_hardware\_params.m Hardware information settings. |

Definition of the coordinate system in the experimental aircraft

The pitch angle is 0 when perpendicular to the ground, and positive when tilted forward. The forks are parallel to the ground (ground reference ) /perpendicular to the pitch angle (device reference ) is 0, and positive when raised up. The fork reference can be determined by " sw\_fcord " (1 - ground, 0 - device).

TBRobot\_main.m

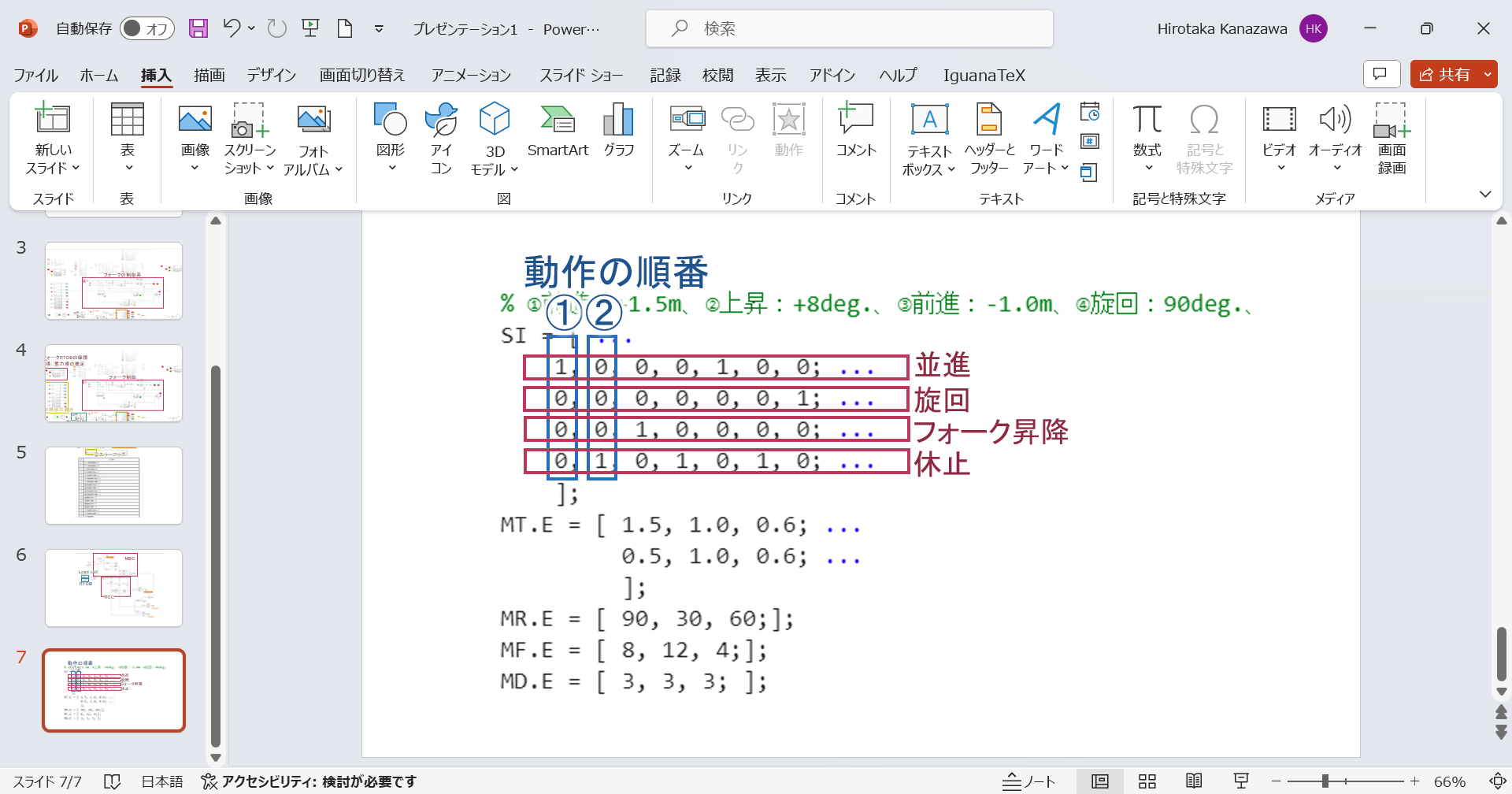
By executing this m file, all of the parameters required for the experiment will be stored in the workspace. There is no need to execute m files in the functions or params folders. This m file mainly sets the ① parameters, ② switches, and ③ operation command values of the controller.

1. Change the parts that correspond to the parameters used in the parameter setting   
   " TBRobot\_exturnal19a\_safe.slx ".
2. Switch Settings:   
   You can experiment under different conditions by simply changing the switch section of TBRobot\_main.m and running it. For details, please check the corresponding section in "   
   TBRobot\_exturnal19a\_safe.slx ". Basically, On is 1 and Off is 0.
3. Movement command values   
   can be generated in the " %% Reference generation " section. The order of movements is defined in the matrix " SI " , and detailed settings such as translation, rotation, fork lifting and lowering, and resting position, speed, and time are set in the matrices " MT.E ", " MR.E ", " MF.E ", and " MD.E ". This can be difficult to understand, so we will explain using an example.

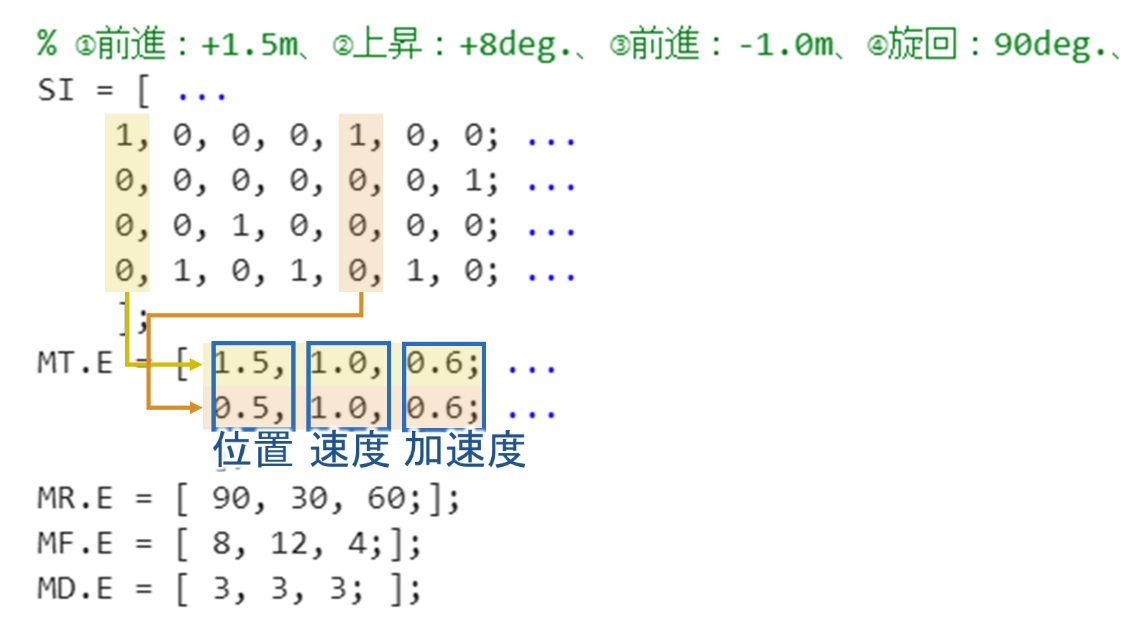
For example, consider a program that moves forward 1.5 m , raises the forks 8 deg. , moves backward 1.0 m , and turns 90 deg.

"SI" is a matrix that defines the order of actions. The vertical axis represents the order, and the horizontal axis represents the type of action. The first row is translation, the second row is rotation, the third row is fork lifting and lowering, and the fourth row is pause.

First, we move forward 1.5 m , so in the first column, we input [1 0 0 0] T so that the movement type is translation . Basically, there will be a pause between movements, so in the second column we input [0 0 0 1] T .

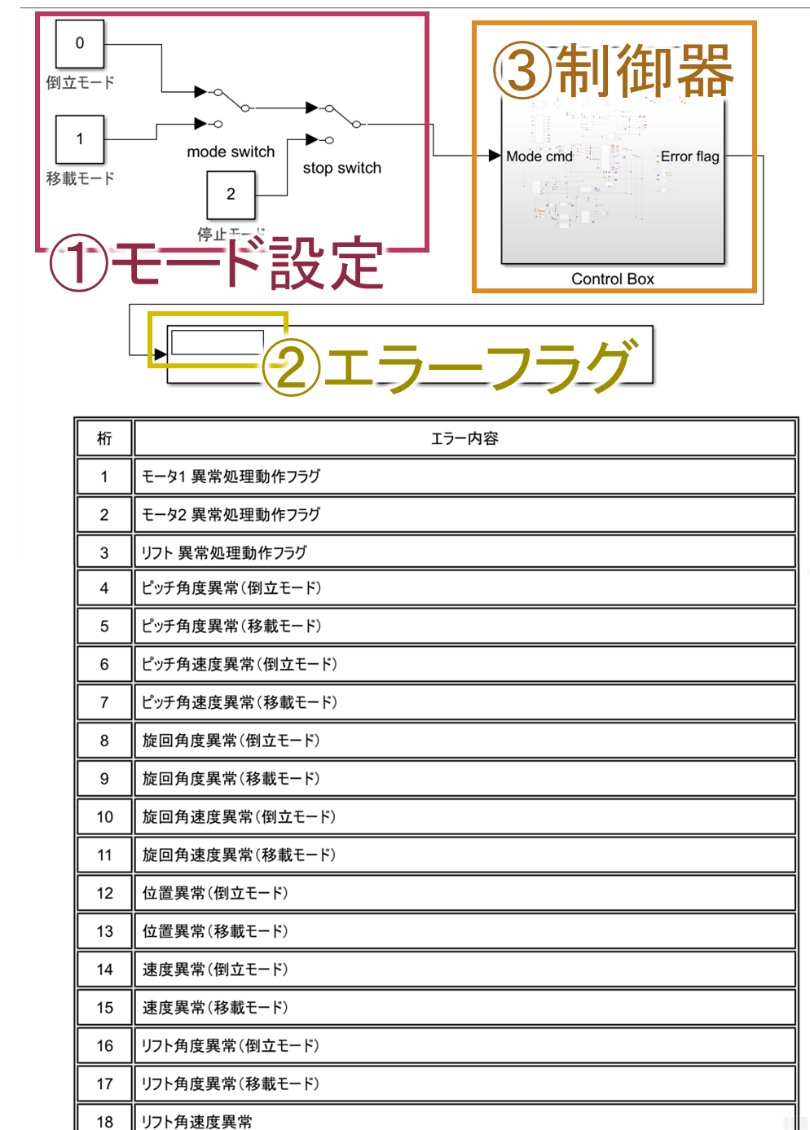


Next, set the details of the action. MT.E is the translation, M RE is the rotation, M FE is the fork, and M DE is the rest setting matrix. MT.E , M RE , and M FE each set the position in the first column, the speed in the second column, and the acceleration in the third column, and move to the position set in the first column at the speed set in the second column. The acceleration in the third column is the acceleration from 0 to the speed set in the   
second column. If the same type of action is performed multiple times in a series of actions, you can do so by adding more rows. In this example, a translation action occurs twice in the series of actions, the first one is forward 1.5 m , and the second one is backward 1.0 m , so the translation setting matrix has two rows. Note that if you move forward 1.5 m and backward 1.0 m , you will be at a position of 0.5 m , so the 2 1 component is 0.5. For M DE only, specify the number of seconds for each rest action in each column . In the example below, there is a 3-second pause between each action.



TBRobot\_exturnal19a\_safe.slx

S I will explain the contents of   
Simulink . T The WinCAT settings are set to execute the program named " TBRobot\_exturnal19a\_safe.slx ", so do not change the name of the Simulink program.



1. Mode setting:   
   Switch between inverted mode, transfer mode, and stop mode using the manual switch . Details will be explained in the experimental procedure.
2. Error Flag   
   Two-wheeled forklifts have various restrictions imposed on them for safety reasons. If they stop due to restrictions, the reason for the stop will be displayed here. For example, if it displays something like "0 000 0000 0000 0001 0000 ", which is the fifth digit, it would be something like "Pitch angle abnormal (loading and unloading mode)". If you want to operate the forklift in a range larger than the normal restrictions, change the settings in " TBRobot\_safety\_params.m ".
3. Controller " Control Box "   
   This is where you design your own controller. When you open it, the following screen will appear.

TBRobot\_exturnal19a\_safe.slx/Control Box



The role of each part is as shown in the image.

RTOB used in the "fork controller" calculates the friction and gravity terms in the outer part (top left). Please refer to the master's theses by Yajima and Kanazawa to find out what kind of calculations are done inside. By the way, since the DOB and RTOB are built on the motor side, please note that the torque actually applied to the fork will not be calculated unless the output RTOB value is multiplied by the gear ratio ( 3/3 for a fork).

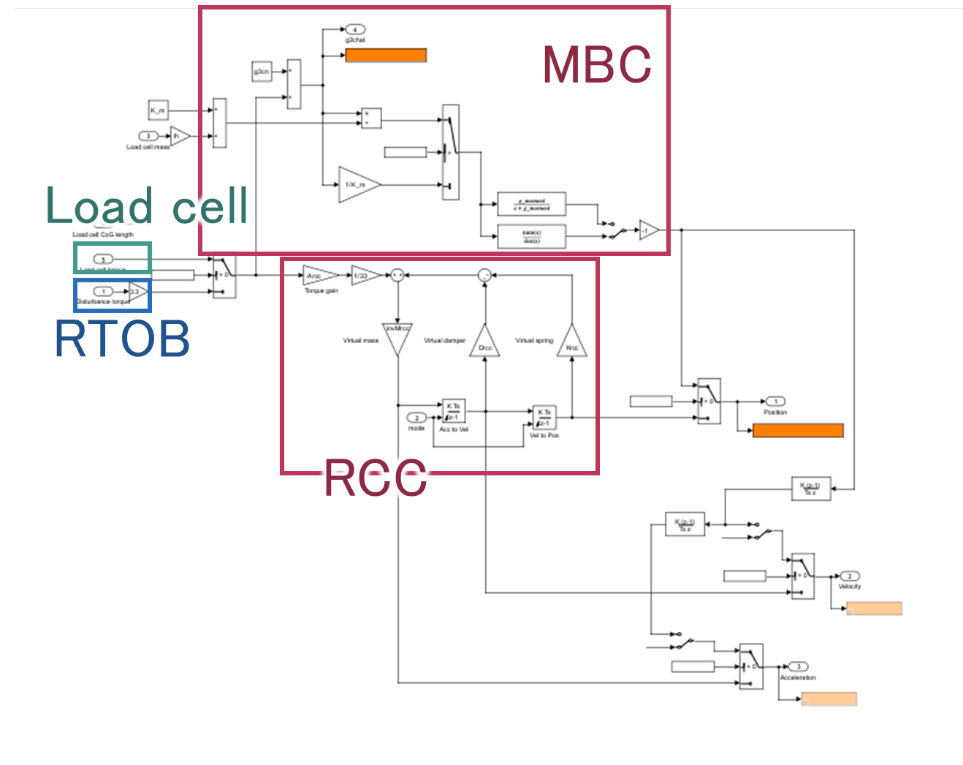
The "fork controller", "swing controller", "S PADO " and "pitch angle P D " are probably not going to be changed often, but if you want to experiment with changing the controller gain or the DOB cutoff frequency, find the parameter values used in Simulink in " TBRobot\_main.m " and change them.

"R CC " include the block diagrams of the R CC used in Yajima's master's thesis and the model-based controller ( MBC) used by Kanazawa in his master's thesis, so you can switch between them to use whichever one you prefer.

The "Kanazawa's proposal" in the black frame contains block diagrams for virtual point calculation, translational speed control using virtual points, and the model error suppression compensator ( MEC) . Translational speed/position control was previously performed by the "Translational position control (old version)" on the right. The "old version" can translate, rotate, raise and lower the forks, and pause. For details, refer to " TBRobot\_main.m " . Translational speed control using virtual points only supports translational speed control. Regarding translational speed control, the old version was made by the loom and is quite a black box, so I personally feel that if you are only performing translational speed control, it is easier to use the version that uses virtual points. The position of the virtual point can also be changed freely.

About the contents of "RCC"

TBRobot\_exturnal19a\_safe.slx/Control Box/Subsystem10/Repulsive compliance controller

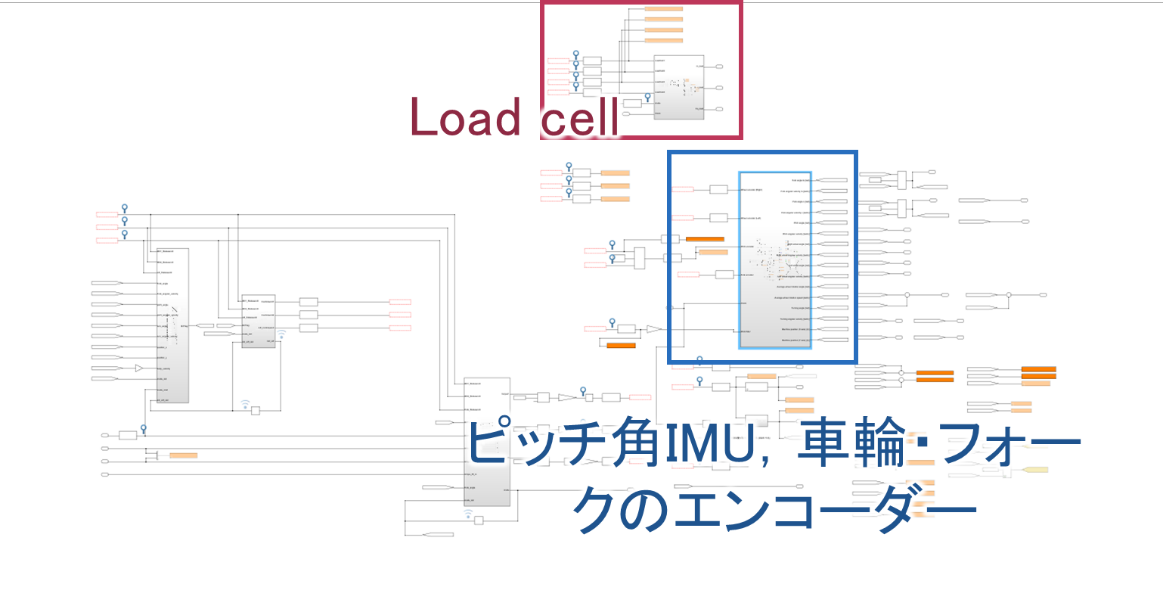


select RCC and MBC , and RTOB and load cell by changing the switch in TBRobot\_main.m .

About the contents of Plant

The Plant part is the part that converts the output (digital) of sensors etc. obtained via TwinCAT into actual values. Basically, you do not need to touch it, but if you add or replace a sensor, you will need to edit this area.

TBRobot\_exturnal19a\_safe.slx/Control Box/Plant



For example, the load cell is connected to a Beckoff "EL3104." The E L3104 digitizes and transmits ± 10 V with a signed 16 -bit resolution, so the received signal is multiplied by to convert it to a voltage, and then the voltage is converted to force according to the load cell catalog.

Acquisition of experimental data

If there is data you want to obtain, you can obtain the data by connecting the "T o File " block.

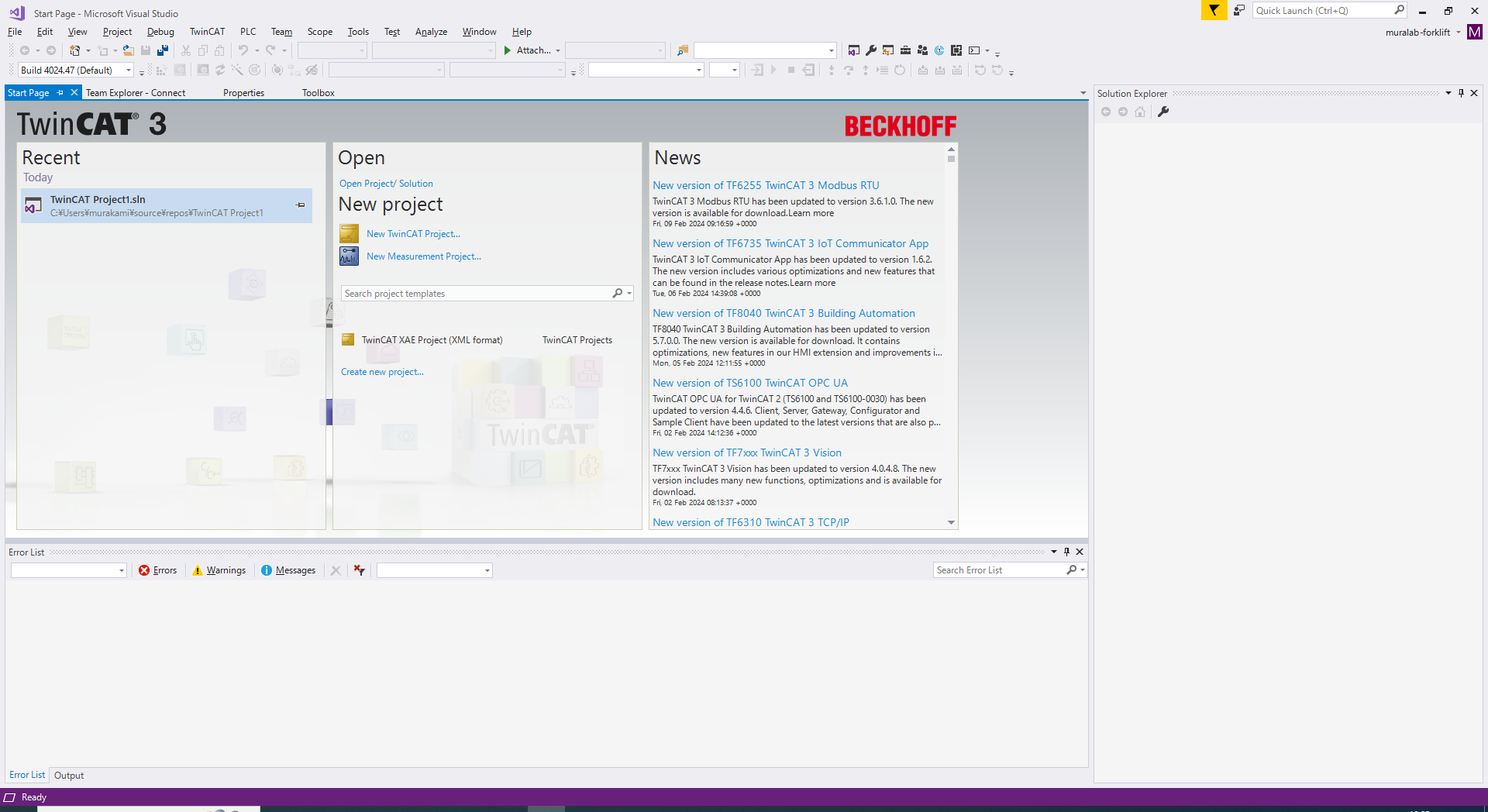
However , the file name must be " C:\share\test\ xxx\_out\_part0 ". It must be in this format when importing the experimental data with tcfile\_converter.m .



TwinCAT

Twin CAT is easy to use by selecting a project that has already been set up.

Select " T winCAT Project1.sln ".



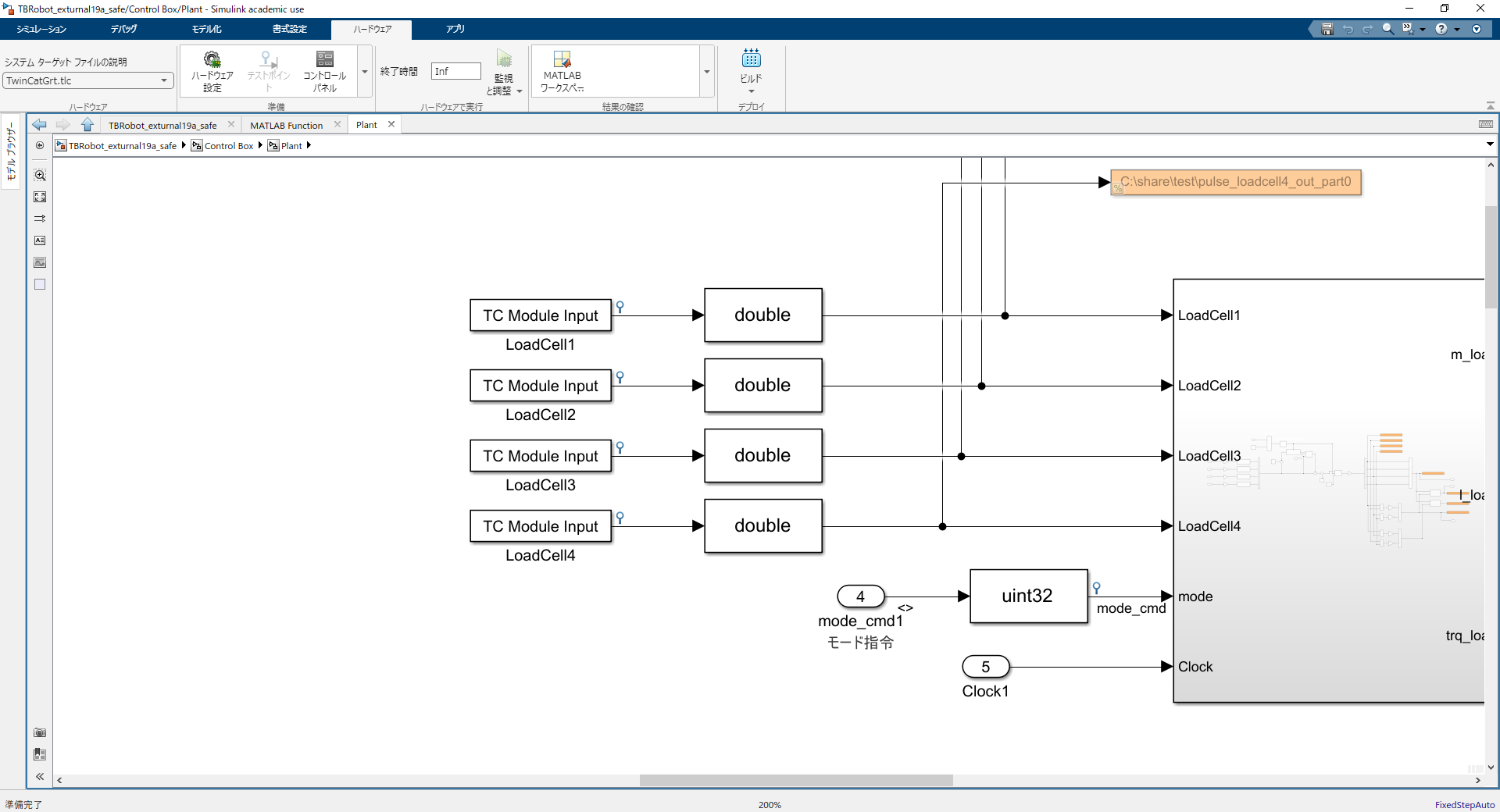
When creating a new project, follow the setup procedure described in "20230526\_TwinCAT Procedure Manual\_MATLAB\_R2022a.xlsx".

About connection with S imulink

In order to reflect the output of sensors, etc. in Simulink , it is necessary to connect using T winCAT .

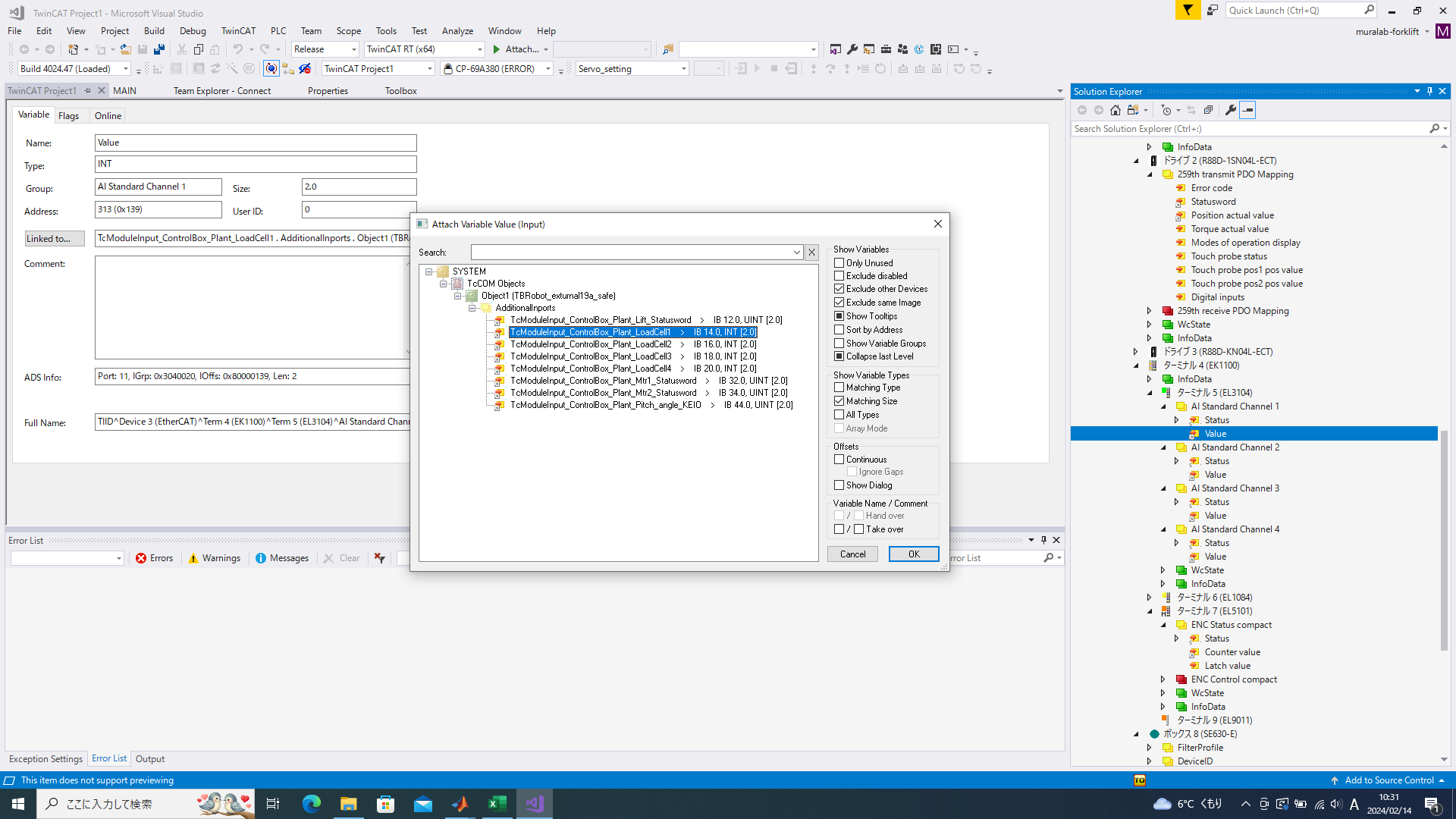
To connect, use Simulink 's "TC Module Input ". Below is an example of a load cell.

↓ Simulink

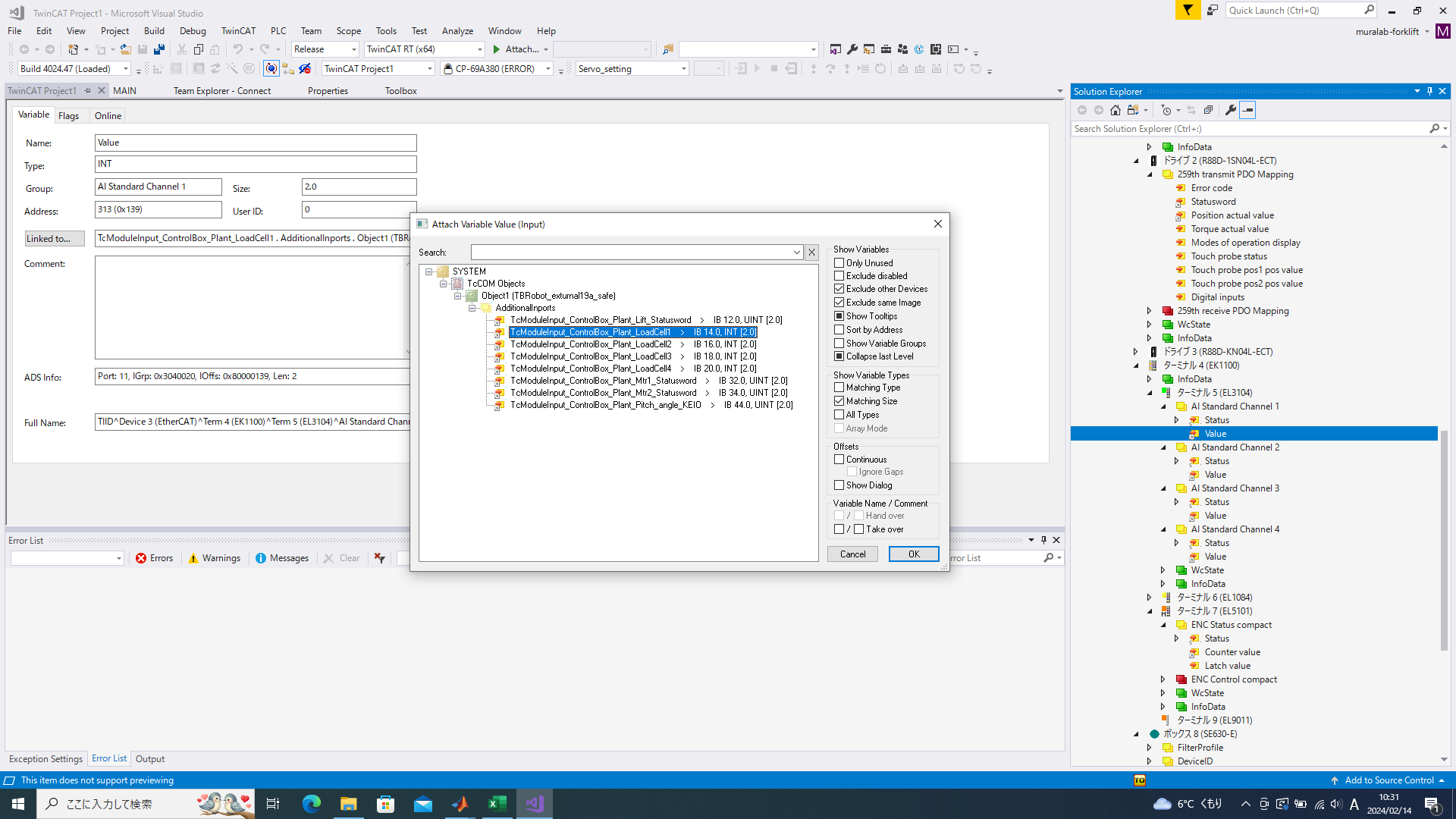


Link to T winCAT according to the variable name of T C Module Input .

1. Select the Value you want to link . (The load cell is connected to the EL3104 .)

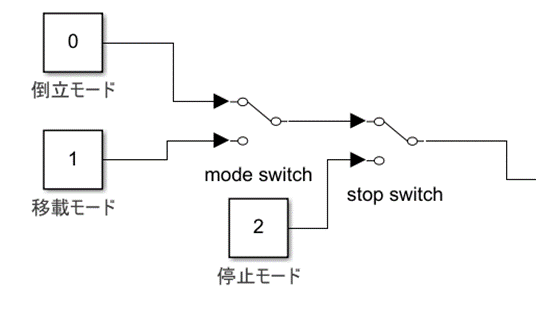
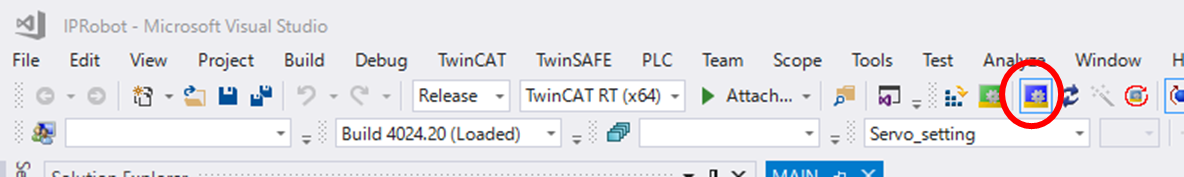
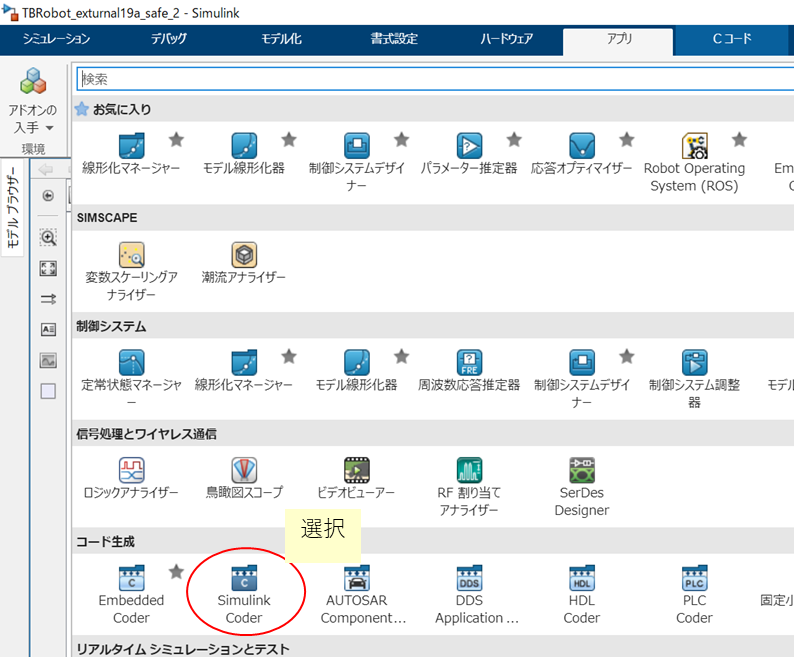
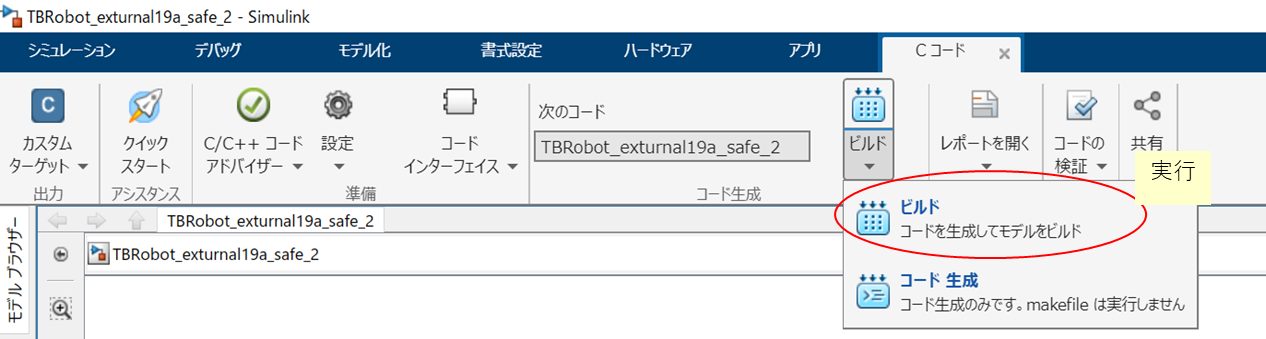
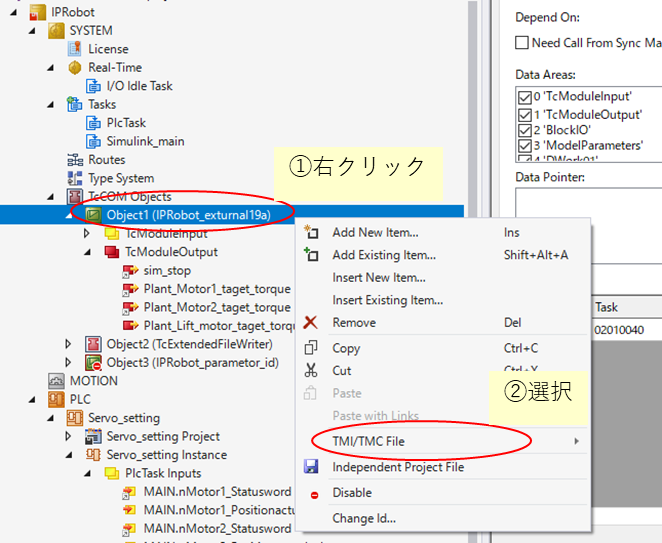
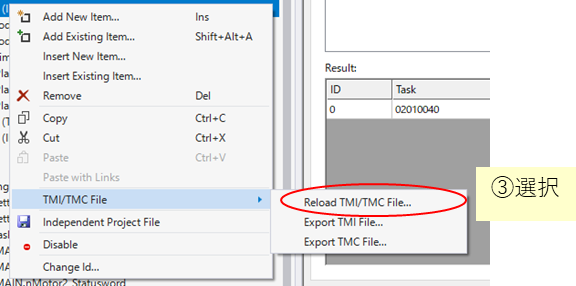
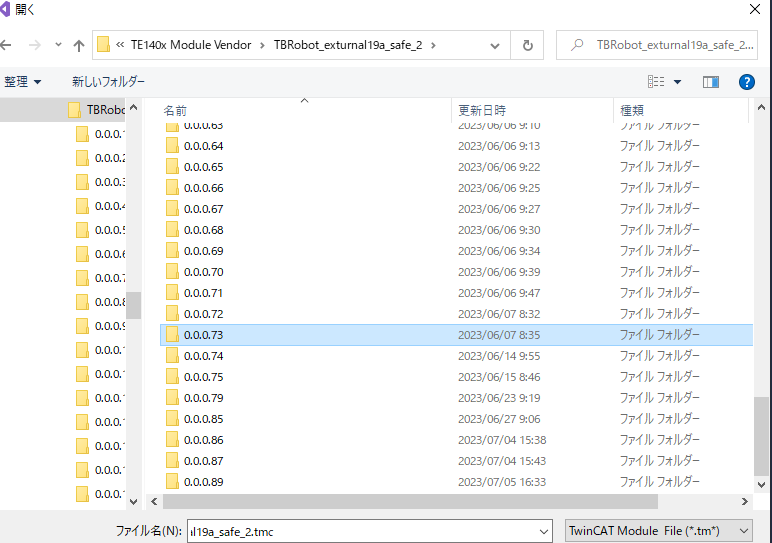


1. the one that corresponds to the variable name on the Simulink side from AdditionalInports . In this case, select "T cModuleInput\_ControlBox\_Plant\_LoadCell1 ". After T cModuleInput \_ are written the Simulink subsystem and variable names.

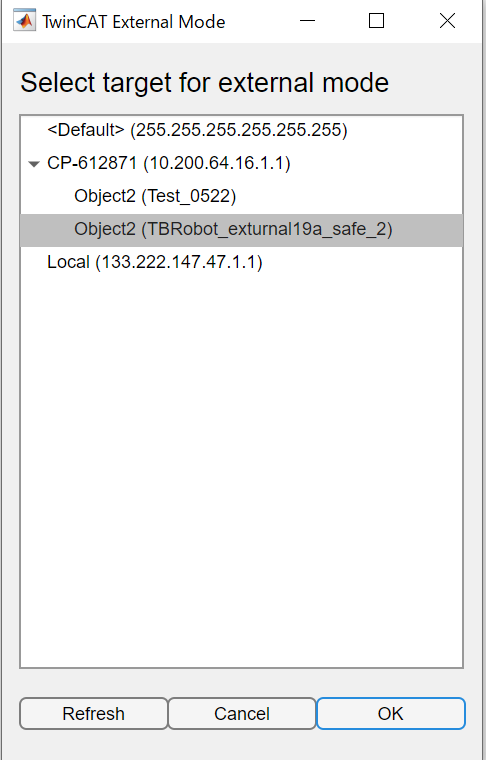
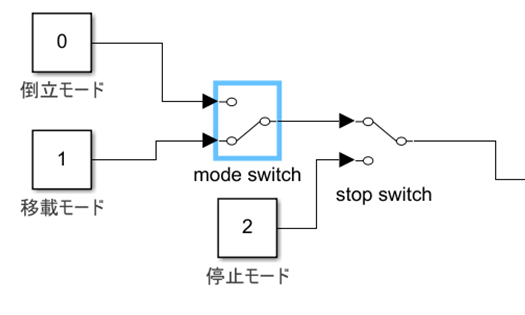
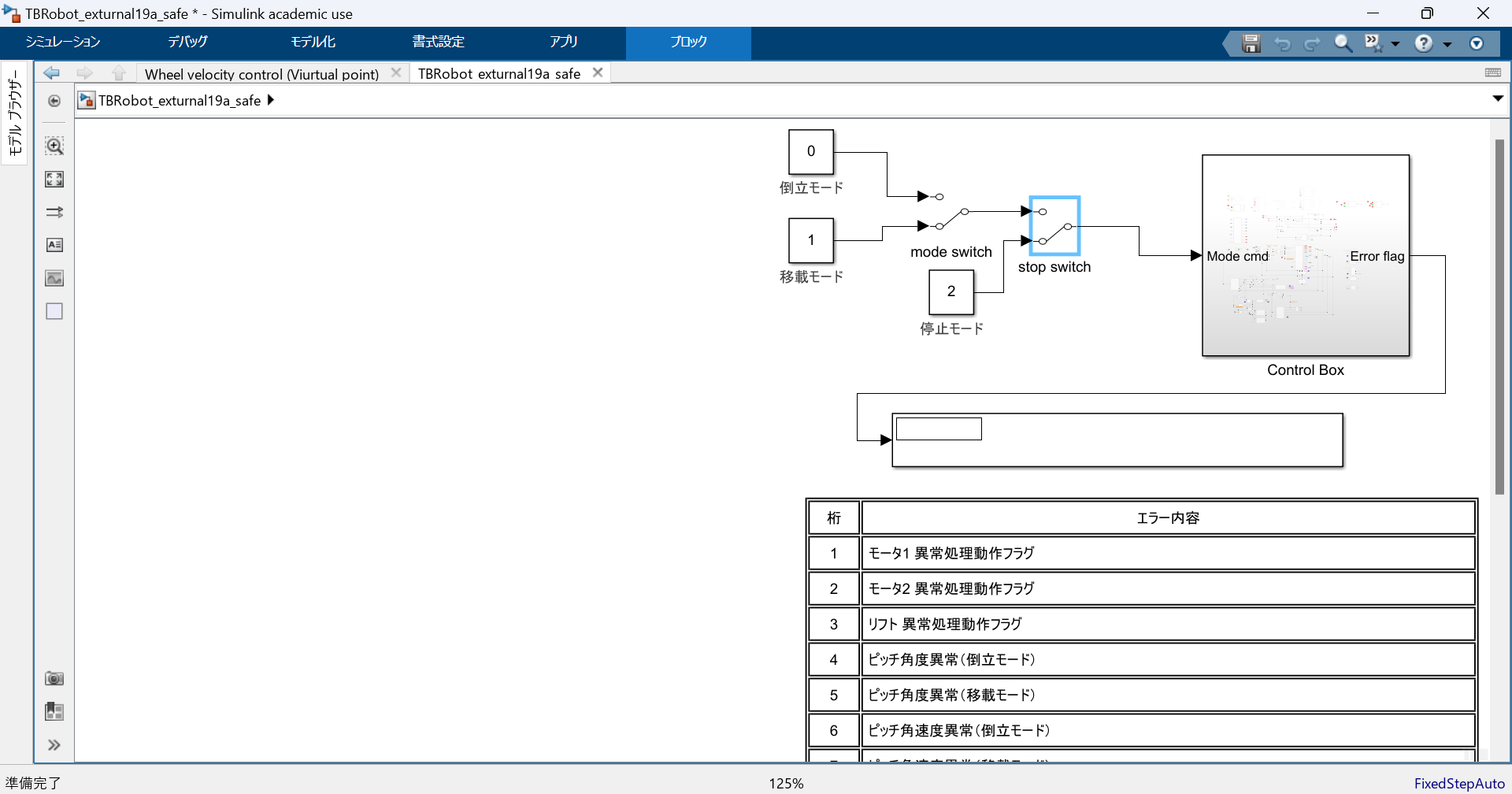
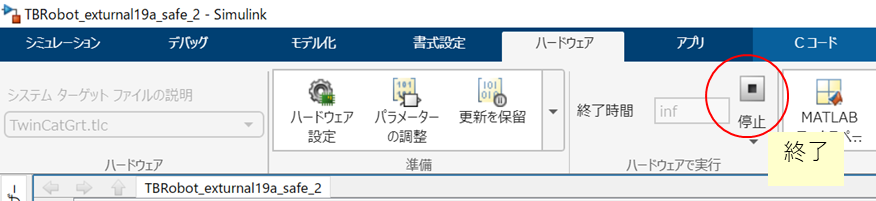


Experimental Procedure

**Preparation**

1. the control box and motor by turning on the main power ( blue button + lever ) .
2. Start MATLAB 20 2 2a and TwinCAT (Visual studio 2017)
3. Simulink mode to "inverted mode" and T winCAT to   
   " config mode".・S imulink  
     
   ・T winCAT  
   
4. " TBRobot\_main.m ", build the model of the desired simulink file ( MATLAB)
5. From the Simulink top tab, select Apps > Simulink Coder  
   
6. Run the build from the C code tab  
   
7. Loading the model built on the TwinCAT side (Visual studio 2017)   
   This operation is required every time you update and build S imulink . ( It is also required when you change the matrix SI that creates the command operation of the m file.)
8. SYSTEM > TcCOM Objects > Right-click on the model you want to update
9. Select TMI/TMC File  
   
10. Select   
    Reload TMI/TMC File...
11. The following window will pop up, so select and open the corresponding file.   
      
    \*A new folder is generated each time you build, and the tmc file will be stored in it, so be careful not to select an older file.  
    
12. Switch to RUN mode ( Visual studio )   
    TwinCAT has a clear distinction between Config mode (blue) for setting and RUN mode (green) for execution. Use RUN mode when experimenting! Config mode otherwise!   
    1. Select the "staircase-like mark" to switch to RUN mode   
    グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, Word

    自動的に生成された説明  
    2. Check the top and bottom of the window to make sure it has switched to   
    グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, Word

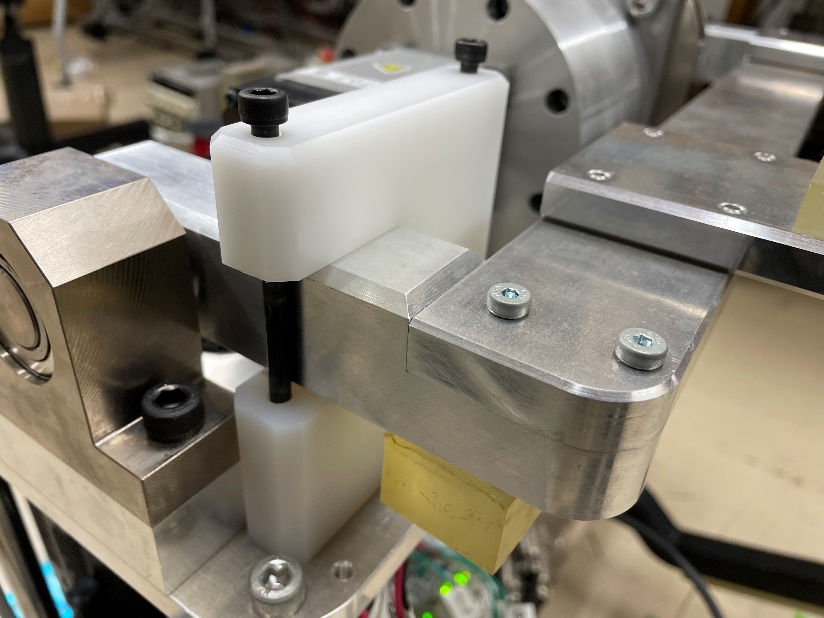
    自動的に生成された説明RUN ( green )
13. Connect to the target in Simulink. ( Matlab )   
    ① Connect to the target from the Hardware > Monitor and Tune > Connect tab in Simulink.   
      
    ② Select the target you want to connect to and click OK.  
    
14. Execute Start   
    Caution: Pressing this will start the machine, so be sure to check for safety!  
    
15. Change to loading/unloading mode (forks lift up)  
    
16. To end the experiment   
    , stop Simulin and press Hardware > Stop   
      
    .
17. Return   
    グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, Word

    自動的に生成された説明TwinCAT to configuration
18. Save the data by running   
    tcfile\_converter.m , entering the name you want to save it as, and then exiting.

When the fork encoder goes wrong

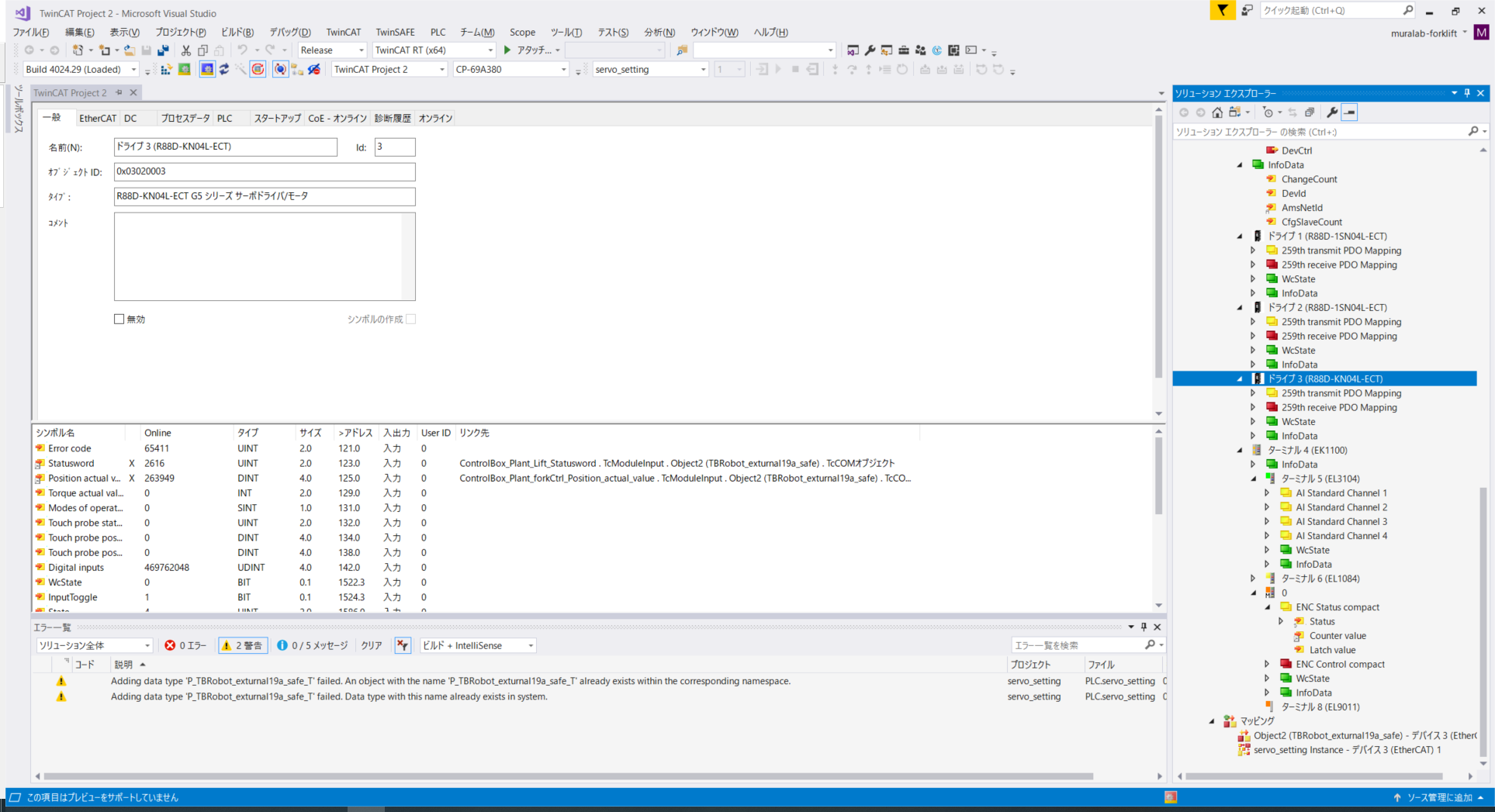
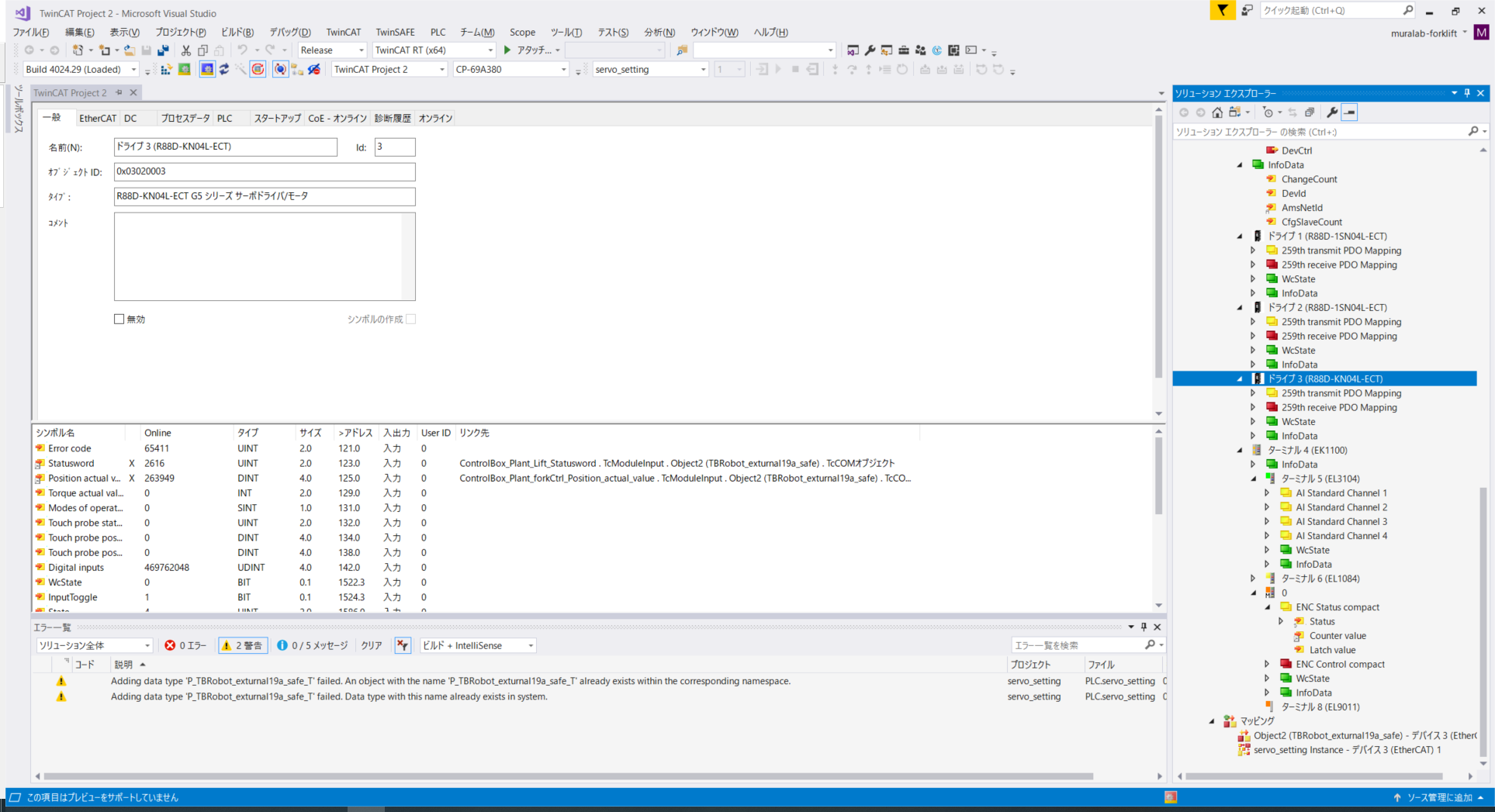
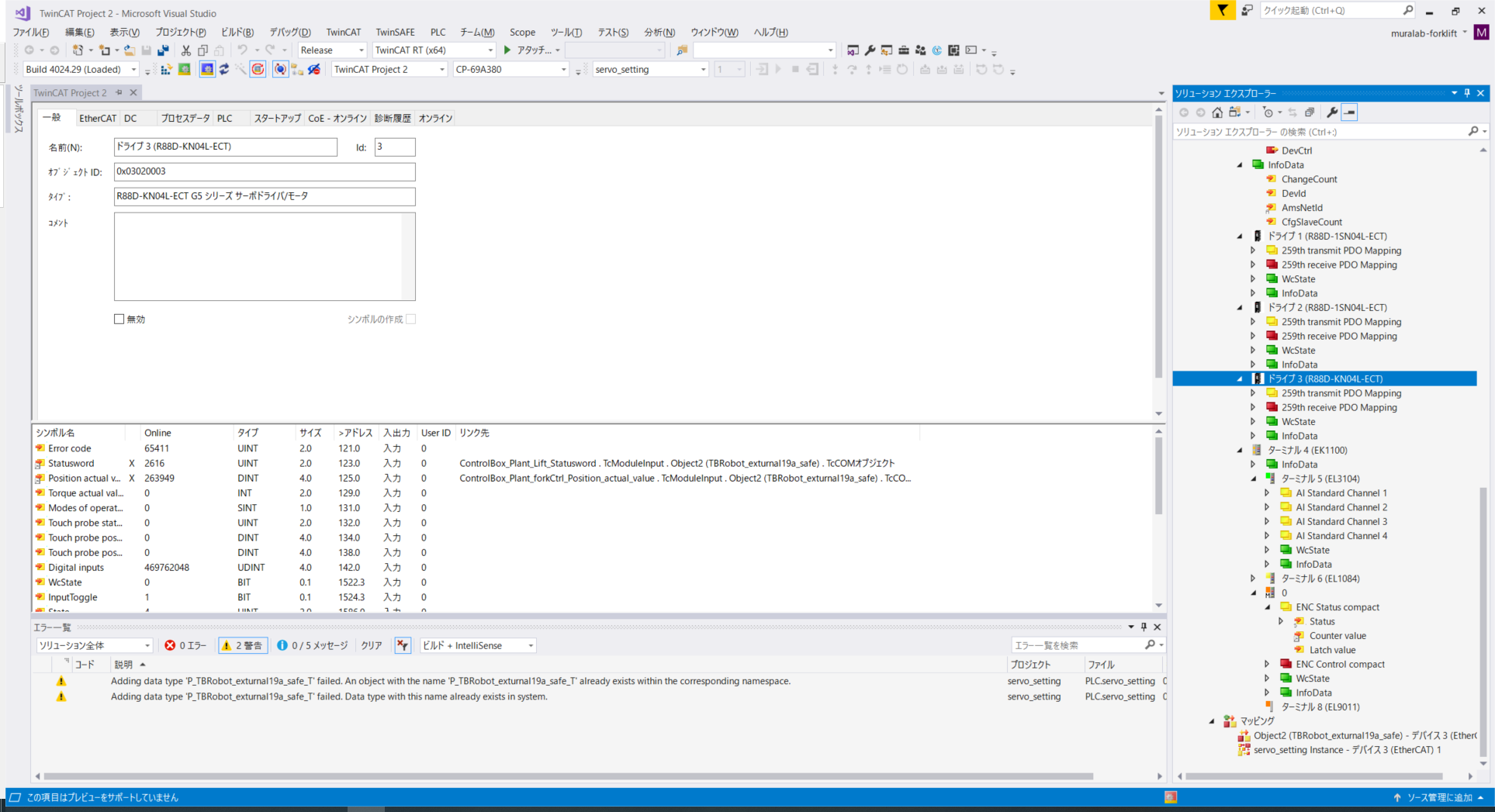
→The fork encoder is absolute, so it may become misaligned if it is subjected to a large impact.

Fix the fork at a 90 degree angle to the body . It is convenient to use the white part (see below).



② Select "Drive 3" of T winCAT (Drive 3 is the input information of the fork encoder)

③ Check the value of " Position actual value "



④ Enter (the value confirmed in ③) x 1.4526 x 10-6 into fork\_offset\_angle in " TBRobot\_hardware\_params.m" .

( 1.4526 × 10-6 is the constant for converting pulses to rad , see the variable p ulse\_2\_forkrad )

