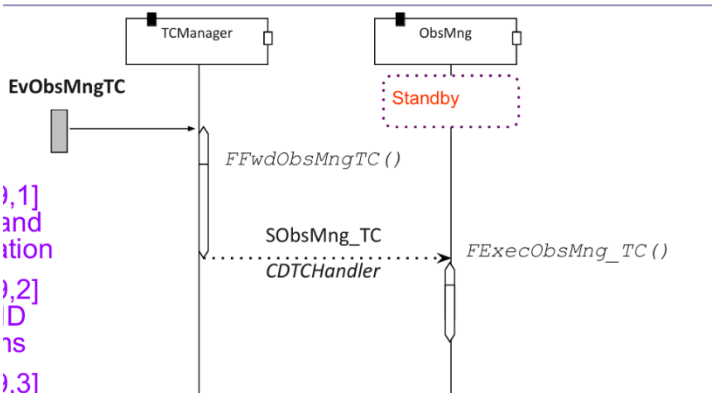
Definición de la clase Protocolo a añadir al Modelo EDROOM, aportando toda la información de cada mensaje:

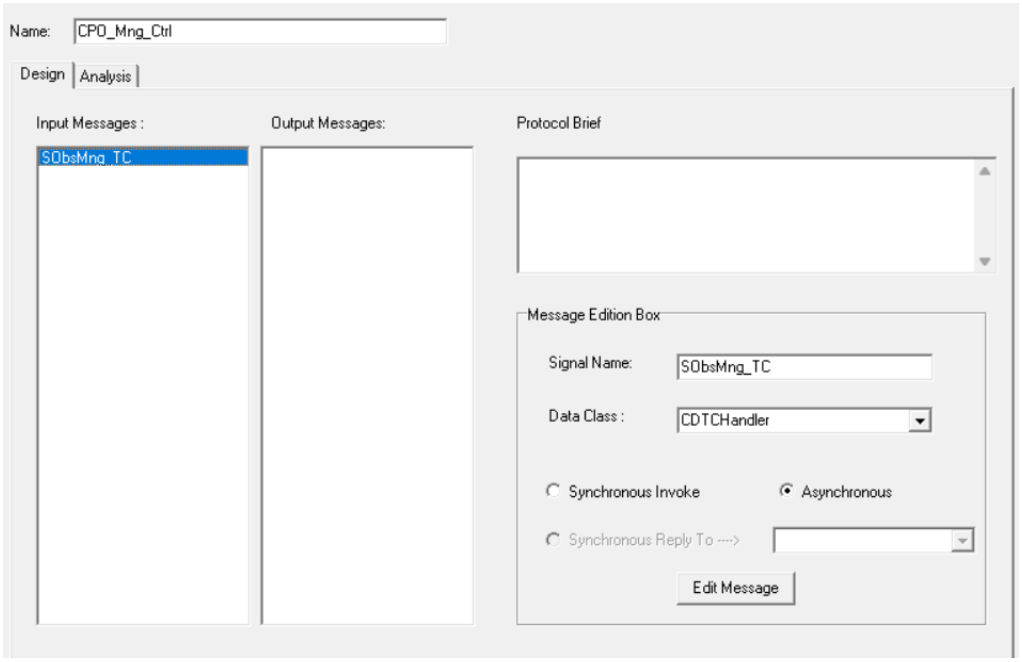
La clase protocolo O_Mng_Ctrl es un puente de comunicación entre el componente TC Manager y ObsMng.



Mensaje de entrada:
SObsMng_TC

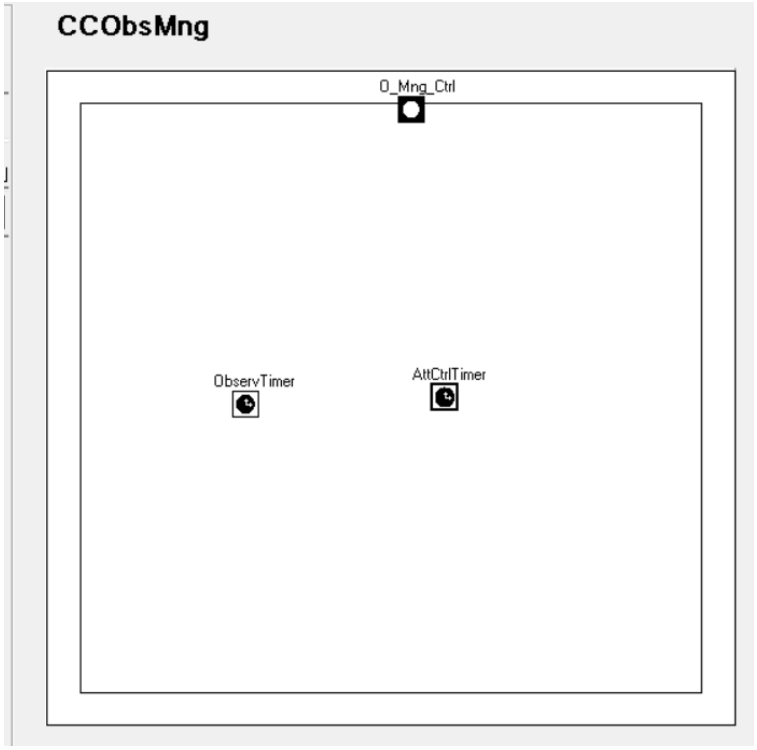
Tipo del dato: CDTCHandler

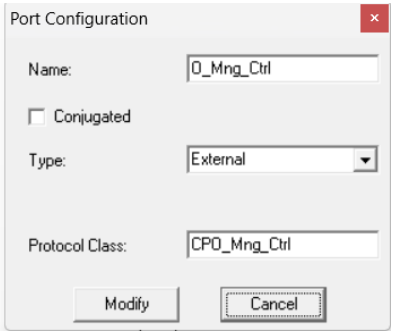
Sin mensaje de salida

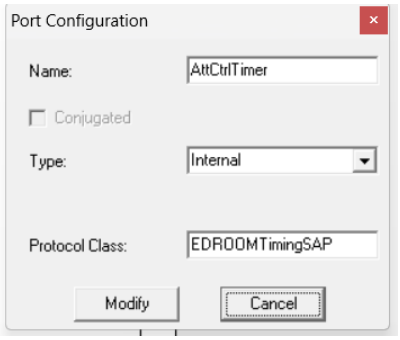


Diseño de la interfaz de la clase componente CCObsMng empleando la misma notación gráfica que se ha proporcionado durante las prácticas, y que debe definir:

- Los puertos de la clase componente, indicando con la notación gráfica correspondiente si el puerto permite solicitar servicios de temporización, o es un puerto de comunicaciones o es un puerto asociado a una Interrupción.

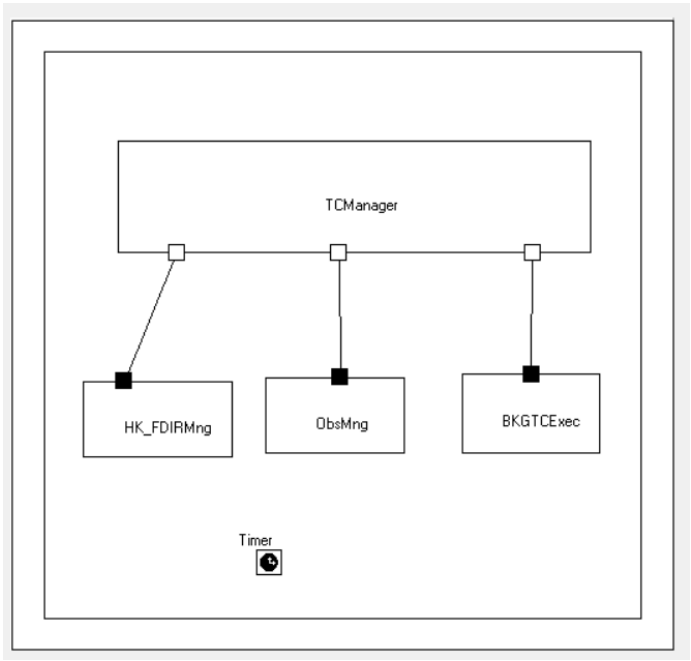


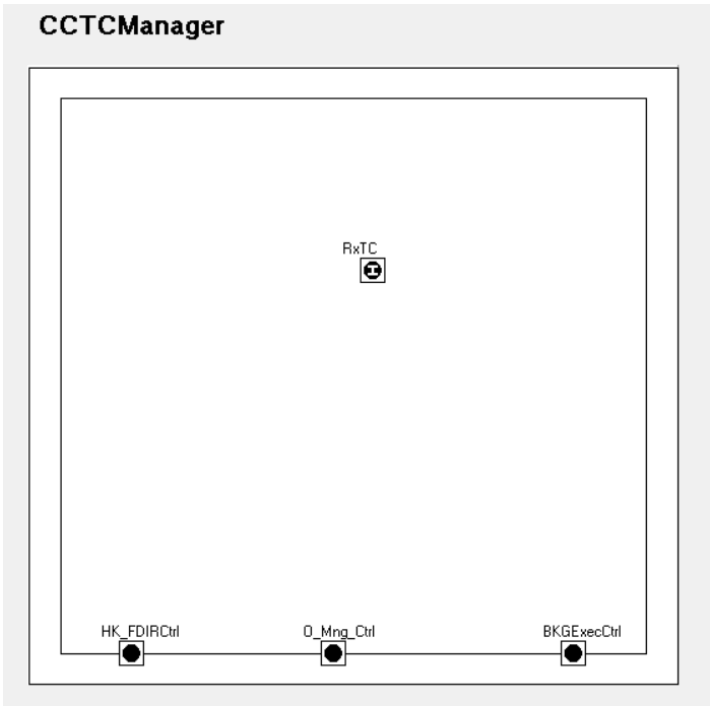




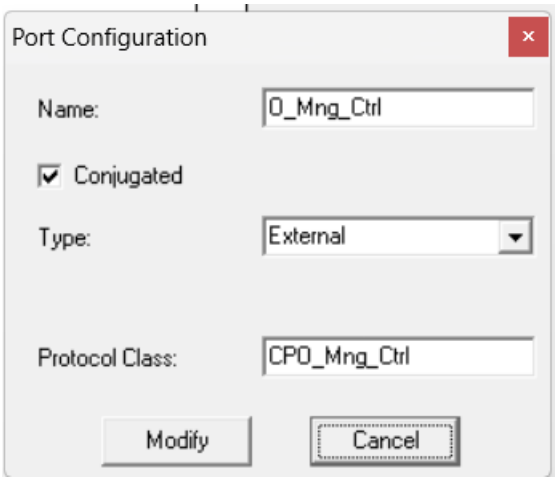
Un puerto de comunicaciones y dos timers (permite servicios de temporización)

- Para los puertos de comunicaciones, indicad la clase protocolo de cada puerto y el tipo de asociación (conjungada o nominal):



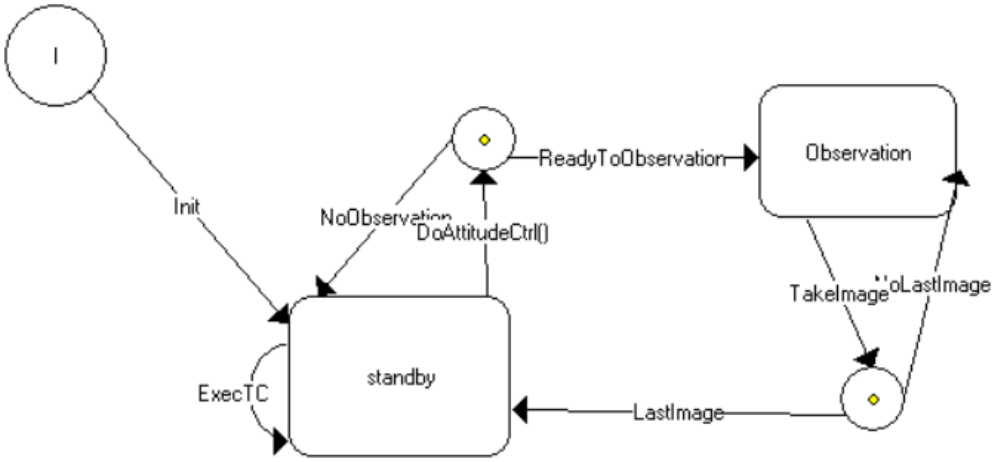


Como vemos tenemos 3 puertos principales que conectan el componente TC Manager con los demás. El nombre que aparece encima de cada puerto es la clase protocolo a la que está asociado. En el componente ObsMng, como hemos visto antes, el puerto es nominal, mientras que en TC Manager los puertos son conjugados:

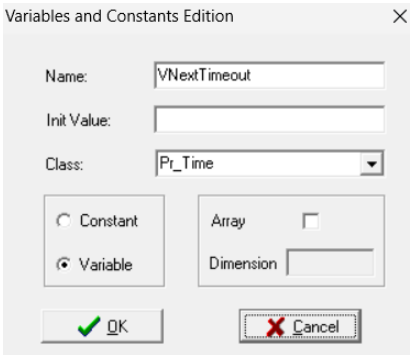


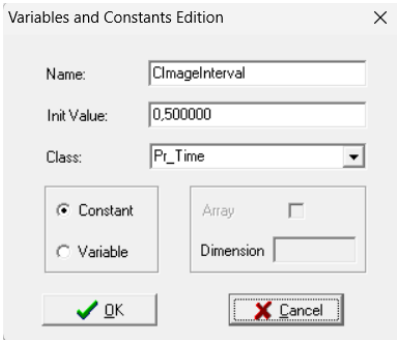
Diseño del comportamiento de la clase componente CCObsMng empleando la misma notación gráfica que se ha proporcionado durante las prácticas, y que debe definir:

- La máquina de estados de la clase componente.



- La declaración de las Variables y constantes de la clase componente.

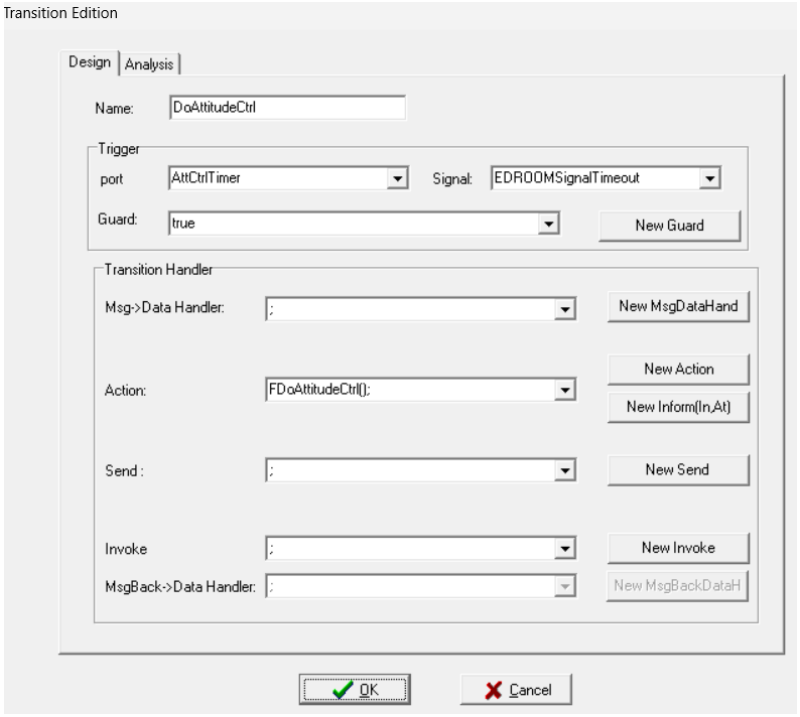


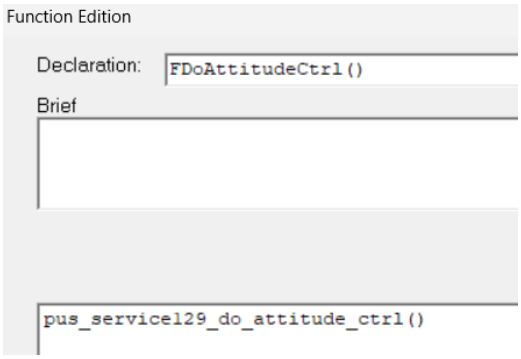


- Definición del trigger de cada transición y la guarda de cada rama

Primera transición: DoAttitudeCtrl

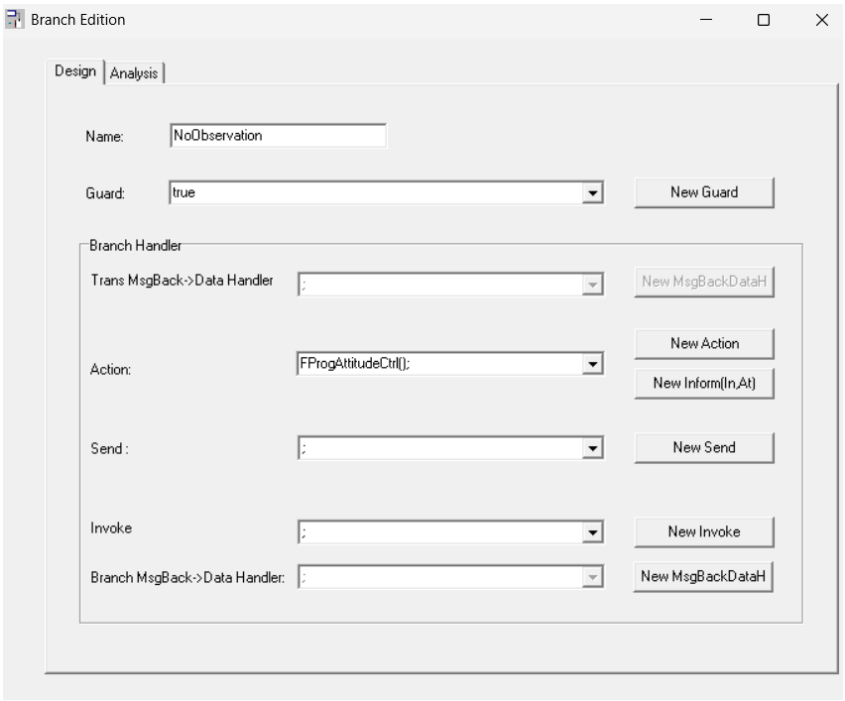
Trigger:

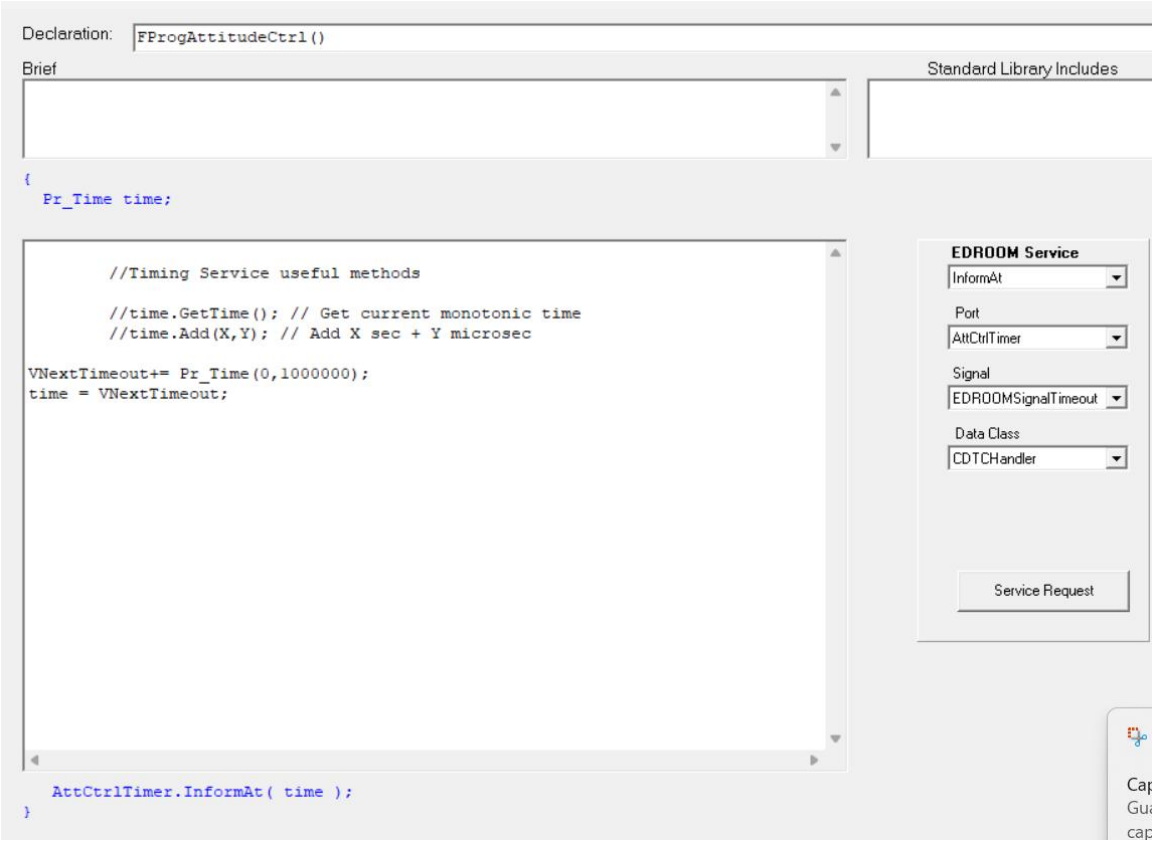




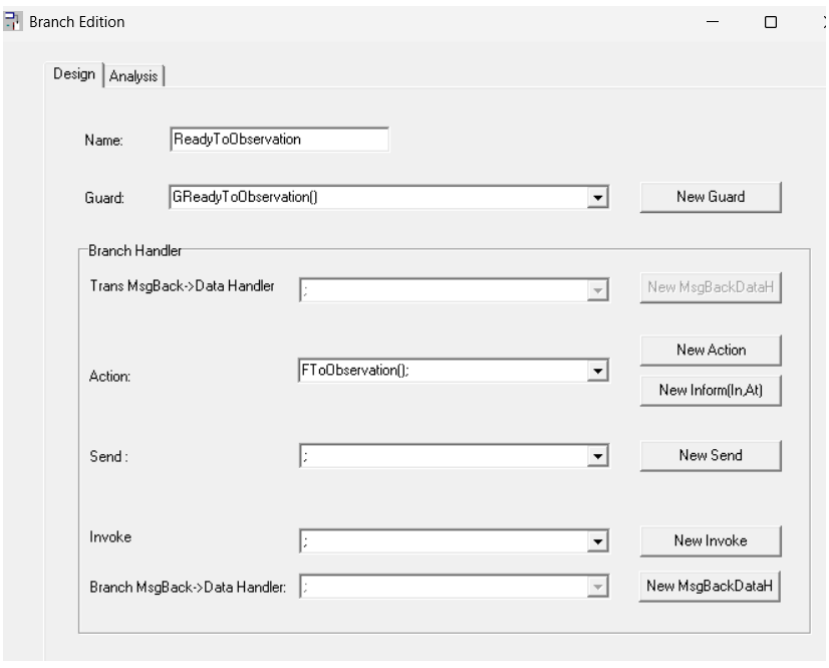
Ramas:

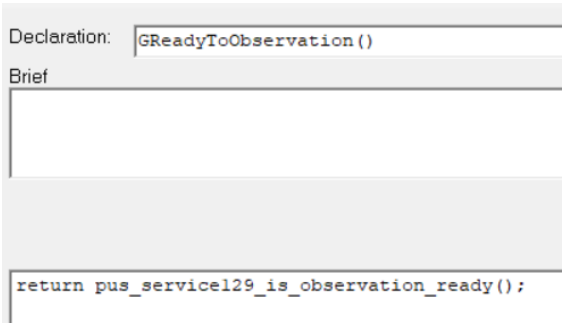
1. No observation:

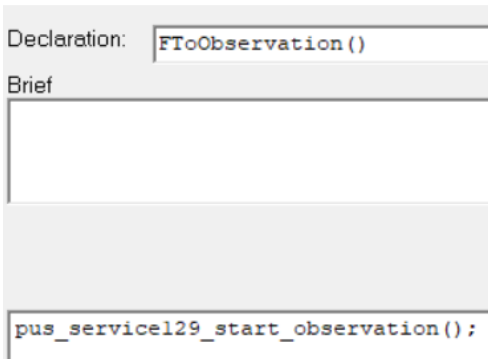




2. Ready to Observation:

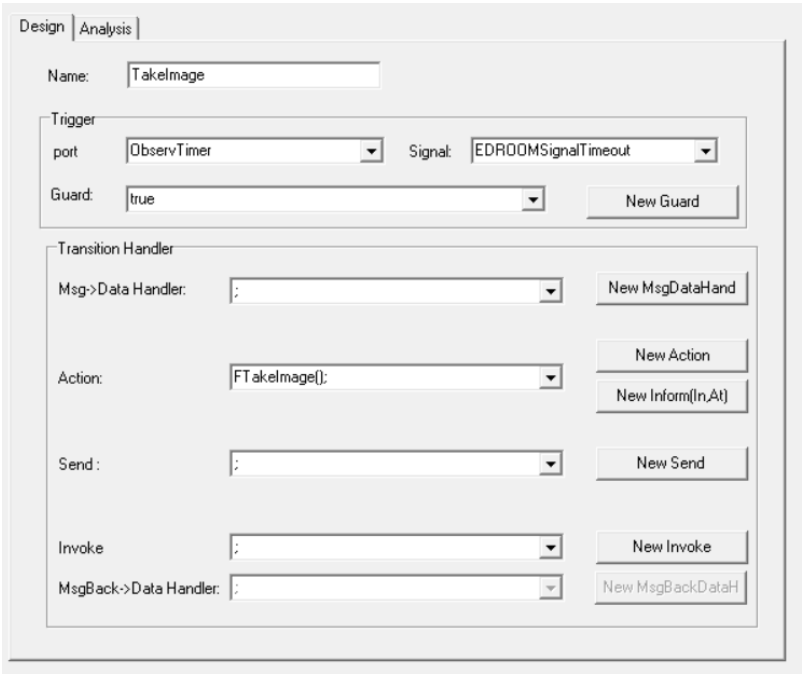


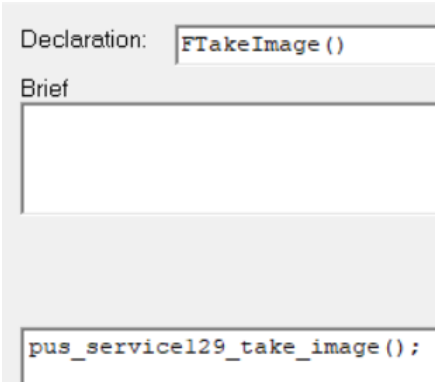




Segunda transición: Take Image

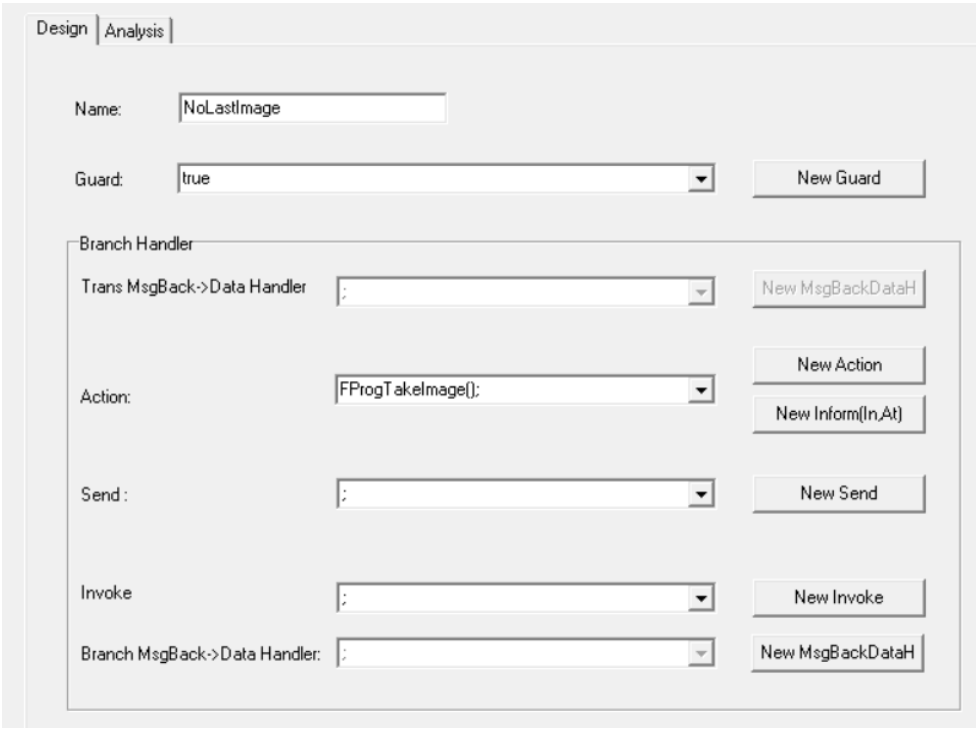
Trigger:

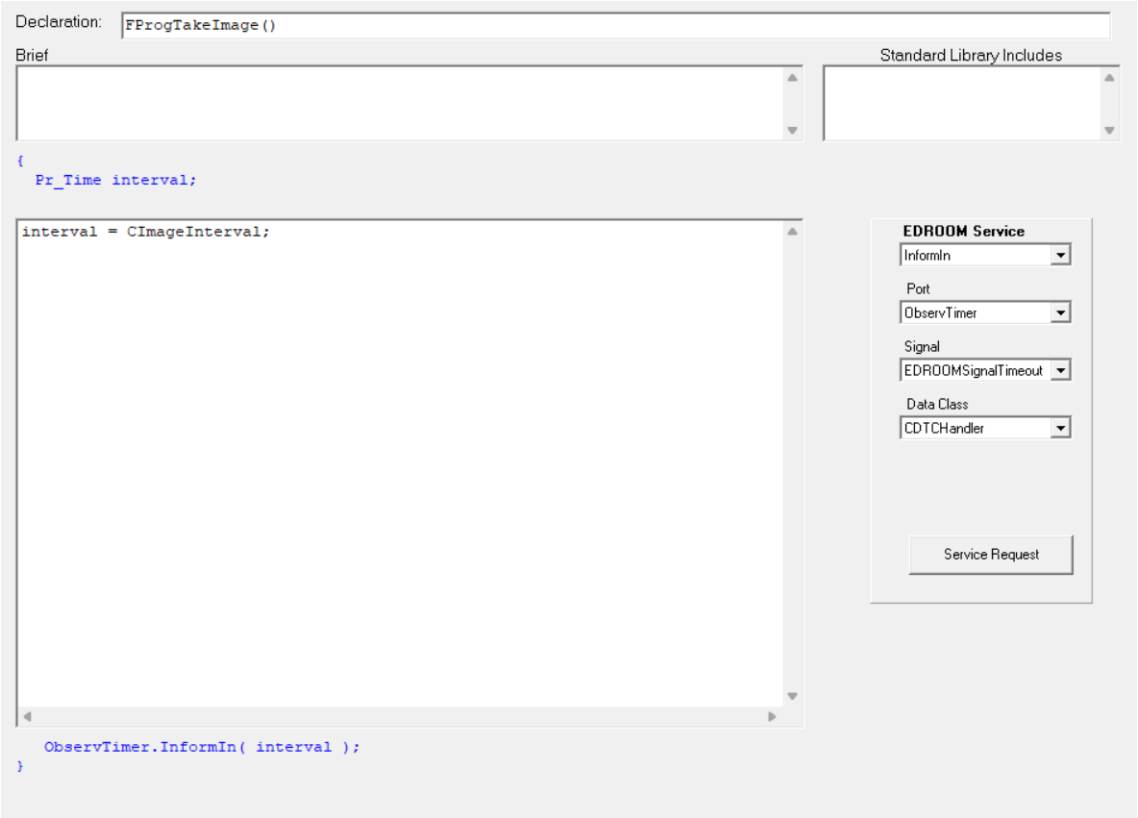




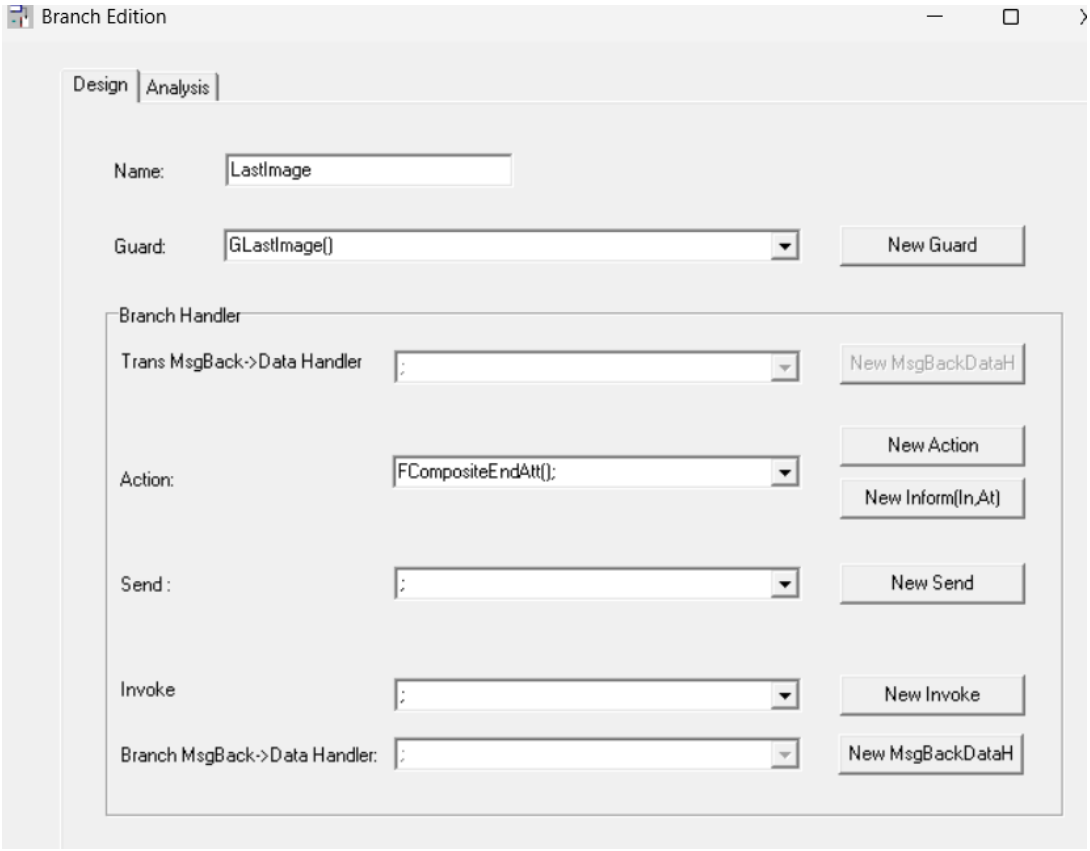
Ramas:

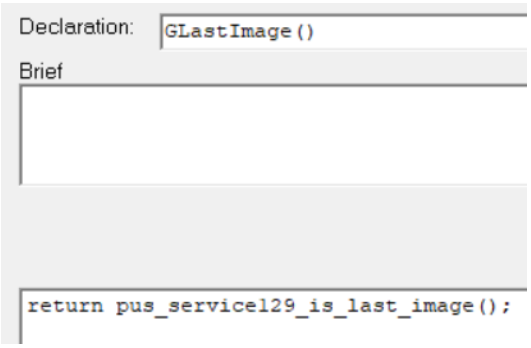
1. No last Image:

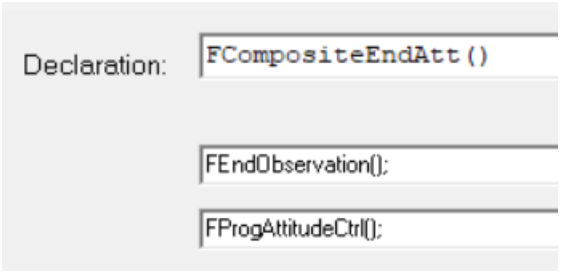



Declaration:
FProgTakeImage()
Brief
Standard Library Includes
{
Pr_Time interval;
interval = CImageInterval;
EDROOM Service
InformIn
Port
ObservTimer
Signal
EDROOMSignalTimeout
Data Class
CDTCHandler
Service Request
ObservTimer.InformIn(interval);
}

2. Last Image:


Branch Edition
Design
Analysis
Name:
LastImage
Guard:
GLastImage()
New Guard
Branch Handler
Trans MsgBack->Data Handler
New MsgBackDataH
Action:
FCompositeEndAtt();
New Action
New Inform(In,At)
Send :
New Send
Invoke
New Invoke
Branch MsgBack->Data Handler:
New MsgBackDataH

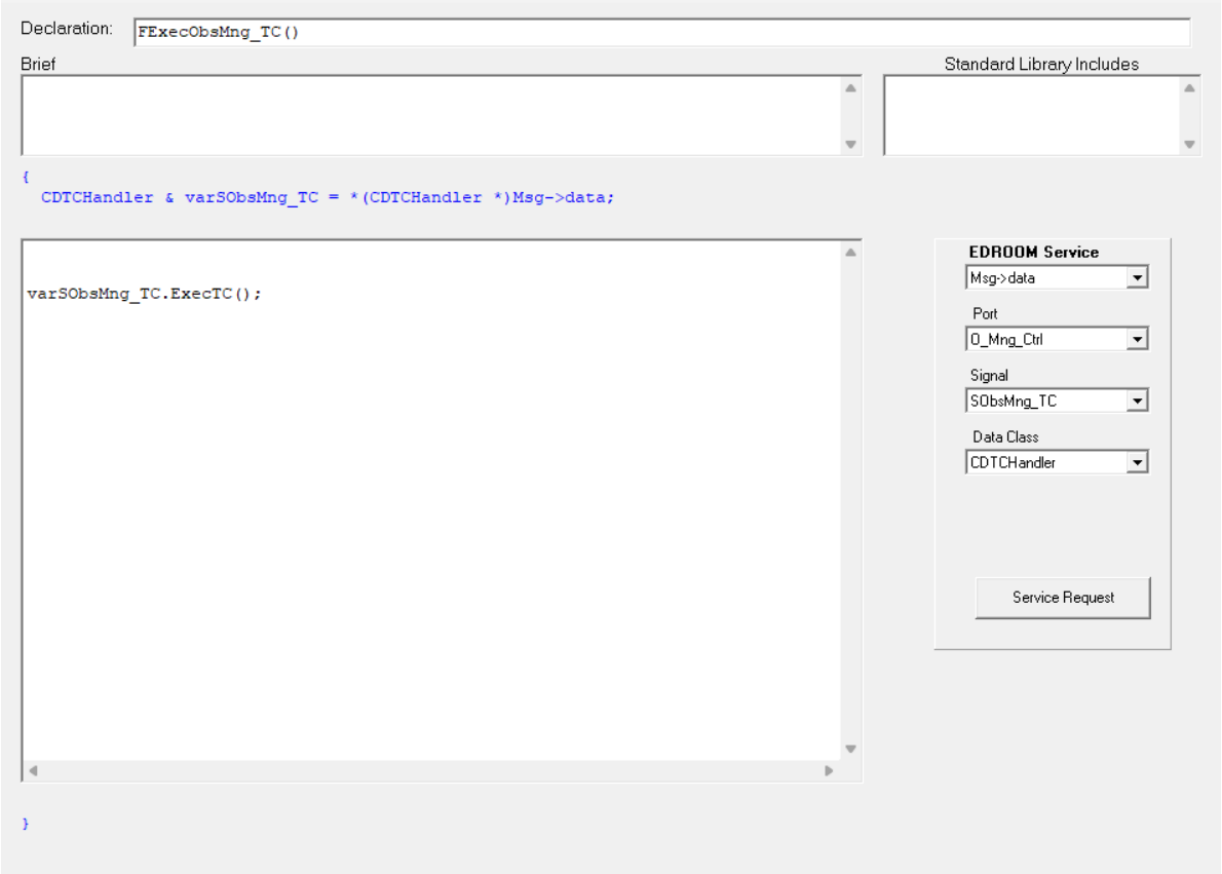




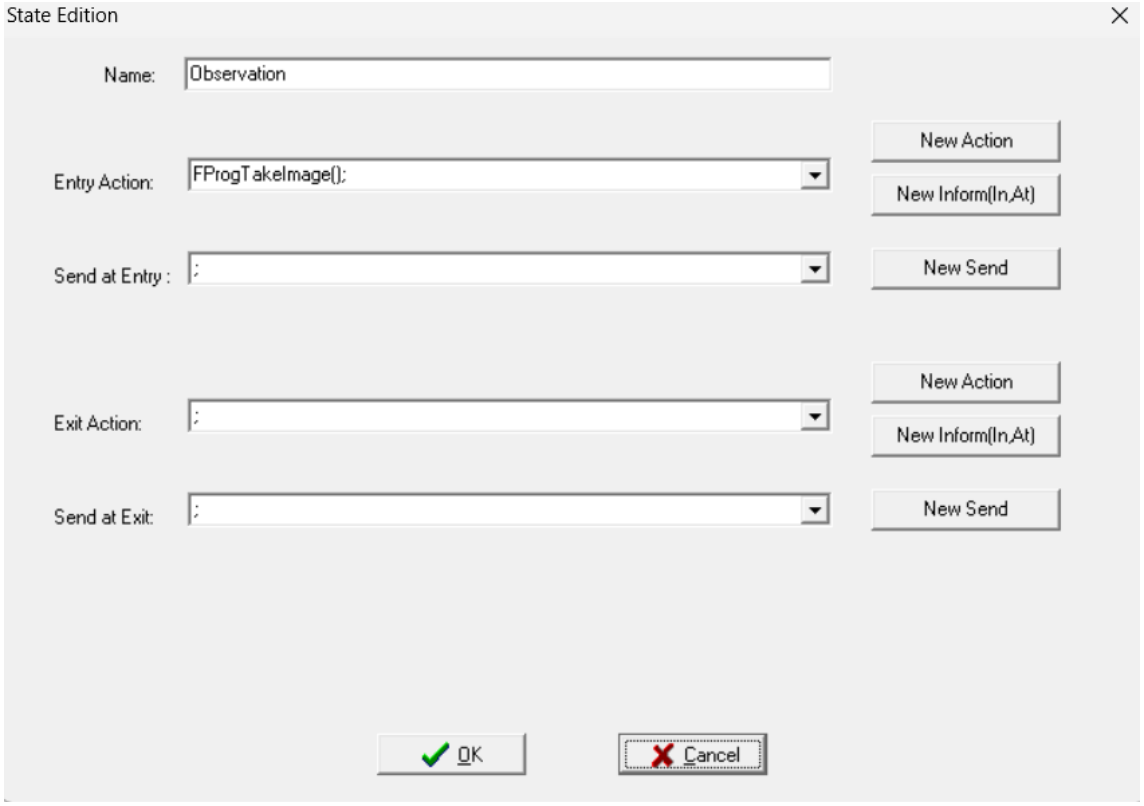
- Actions a ejecutar (teniendo en cuenta que ya hemos visto varias de ellas):

FInit → asociada a una transición, de tipo action

ExecTC → asociada a una transición, de tipo MsgDataHandler



FProgTakeImage → asociada a la entrada de un estado, de tipo action



FEndObservation → asociada a una transición, de tipo action