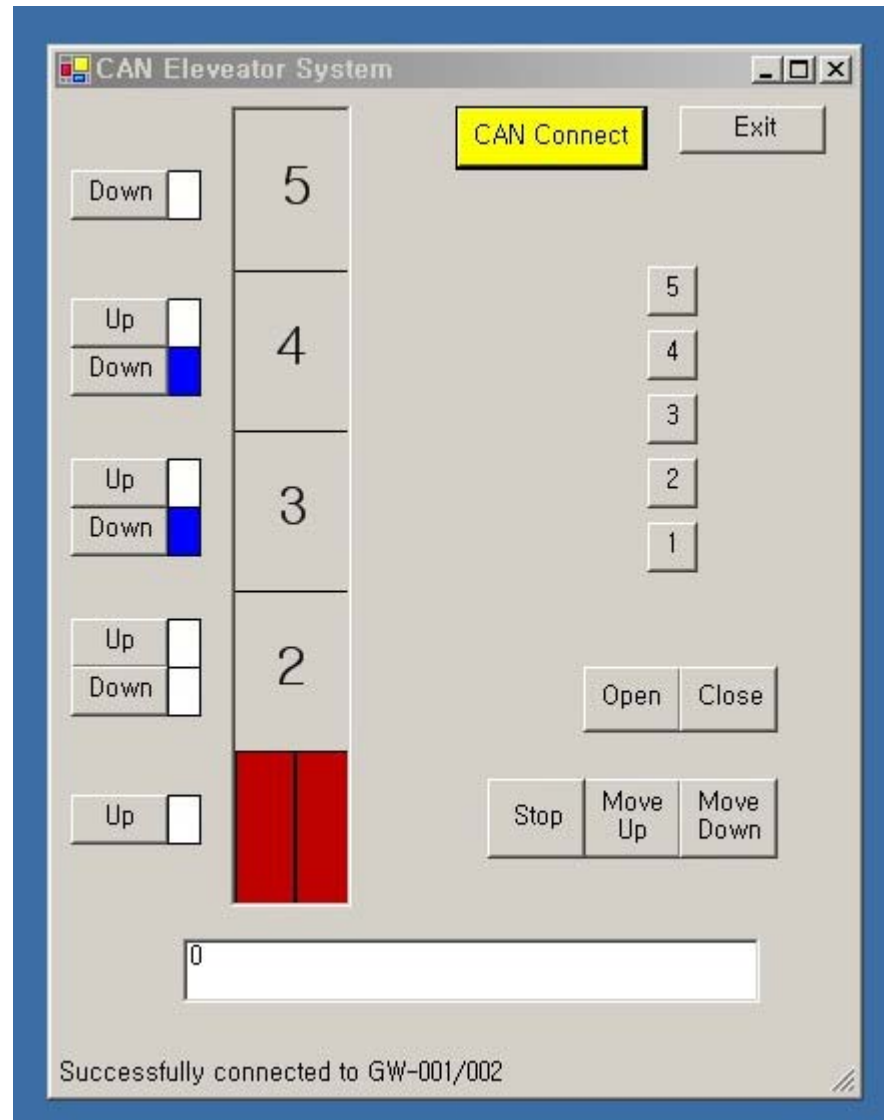


# Design Project

Elevator Controller

CAN, Tasking, and Rhapsody

# Controller for CAN Elevator



# CAN ID & Data

- Rx message ID=1
- Moving up=01,01
- Moving down=01,02
- Stop moving=01,00
  
- Door open=02,01
- Door close=02,02

# CAN ID & Data

- Tx message ID=2
- When the car is moving, a CAN message with the current position in pixel is transmitted 5 times/sec.
- 1<sup>st</sup> byte=1, 2<sup>nd</sup> byte=high byte of position, 3<sup>rd</sup> byte=low byte of position
- Each floor height is 80 pixels
- At the first floor, position is 0. At the 5<sup>th</sup> floor, position is 320.

# CAN ID & Data

- Tx message ID=2
- When a call button is pressed, a CAN message is transmitted
- 1<sup>st</sup> byte=2, 2<sup>nd</sup> byte=floor, 3<sup>rd</sup> byte=1 for up & 2 for down. For example when the 4<sup>th</sup> floor up button is pressed, the data is 2,4,1

# CAN ID & Data

- Rx message ID=1
- When a CAN message with 1<sup>st</sup> byte=3 is received, a call button is off.
- 1<sup>st</sup> byte=3, 2<sup>nd</sup> byte=floor, 3<sup>rd</sup> byte=1 for up button & 2 for down button. For example when a CAN message with the data 3,4,1 is received 4<sup>th</sup> floor up button is off.

# Design Problem

- Design an elevator controller.
- You have to set up your own design objectives to achieve. i.e. the level of complexity for the control algorithm.
- After the design, analyze your own design to determine if your design satisfies your own objectives.

# Design Assumption

- To simplify the problem, the following is assumed: the car is waiting at the 1<sup>st</sup> floor. Everybody wants to go down to the 1<sup>st</sup> floor for lunch.
- Move up to the floor, turn off the call button lamp, open the door, closed the door, go down to the destination floor, open the door, close the door, etc.



# Requirements for Design

- Reliability
- The controller must be able to respond to all the possible key inputs in the reliable and reasonable manner.

# 보고서 제출 요령

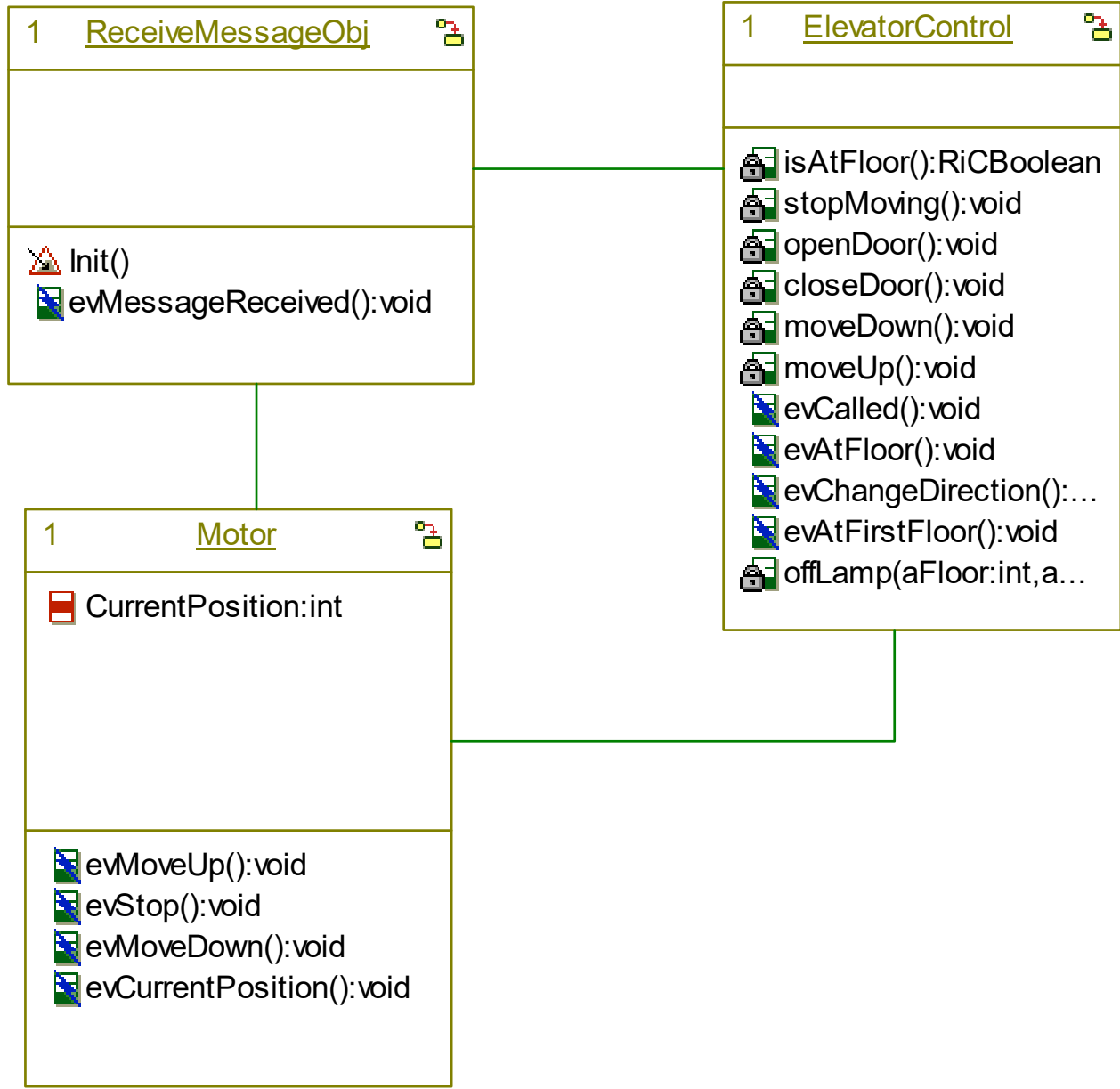
- 보고서와 함께 프로젝트 파일도 e-mail 로 제출할 것
- 빈 directory를 만든 후, Rhapsody project가 열린 상태에서 File menu의 Save As를 이용하여 빈 directory에 저장한다.
- 프로젝트 파일이 저장된 directory를 zip file로 압축하여 [limdj@hanyang.ac.kr](mailto:limdj@hanyang.ac.kr)로 제출

# 보고서 내용

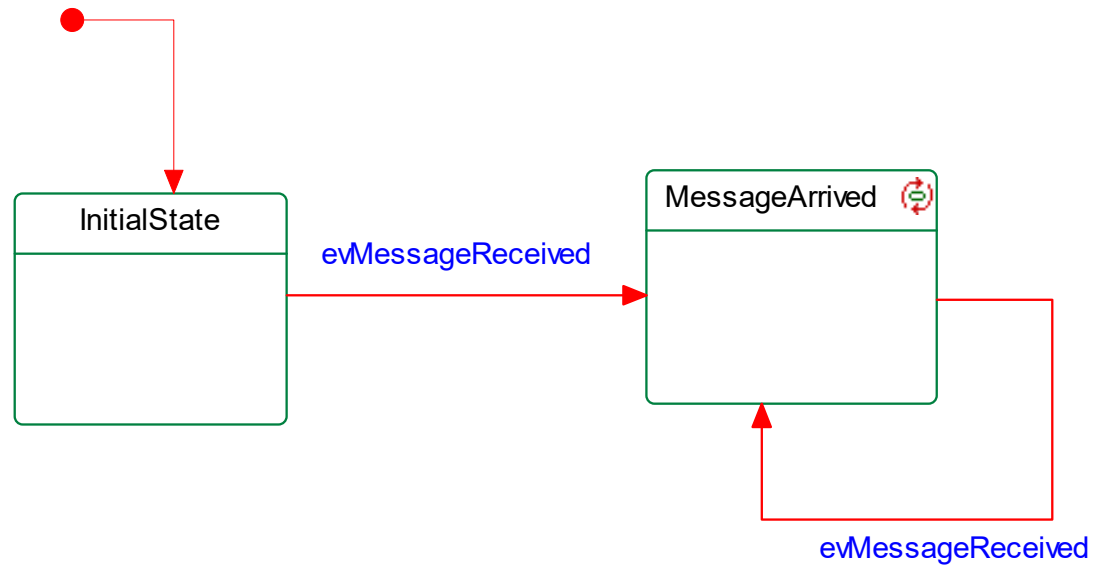
- 본인이 설계하기로 설정한 제어기 동작 시 나리오를 기술
- 본인이 설계한 제어기가 본인이 설정한 기준을 만족하는지 분석한 결과
- 본인이 설계한 제어기의 실제 동작 결과에 대한 기술
- 결과에 대한 결론 및 토의

# 평가

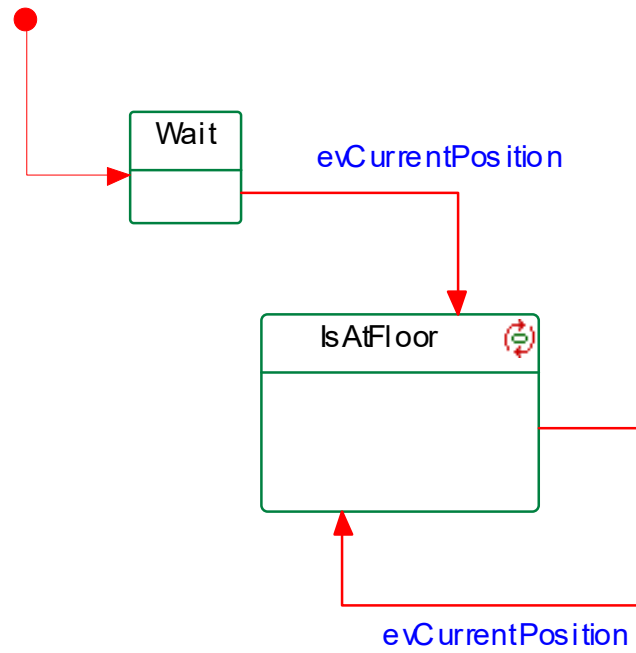
- 기능 구현: 40%
- 설계의 적절성 및 안전성: 30%
- 결과 보고서: 30%
- 본인이 설계하기로 설정한 시나리오의 완성도에 따라서 가점 또는 감점이 있을 수 있음.



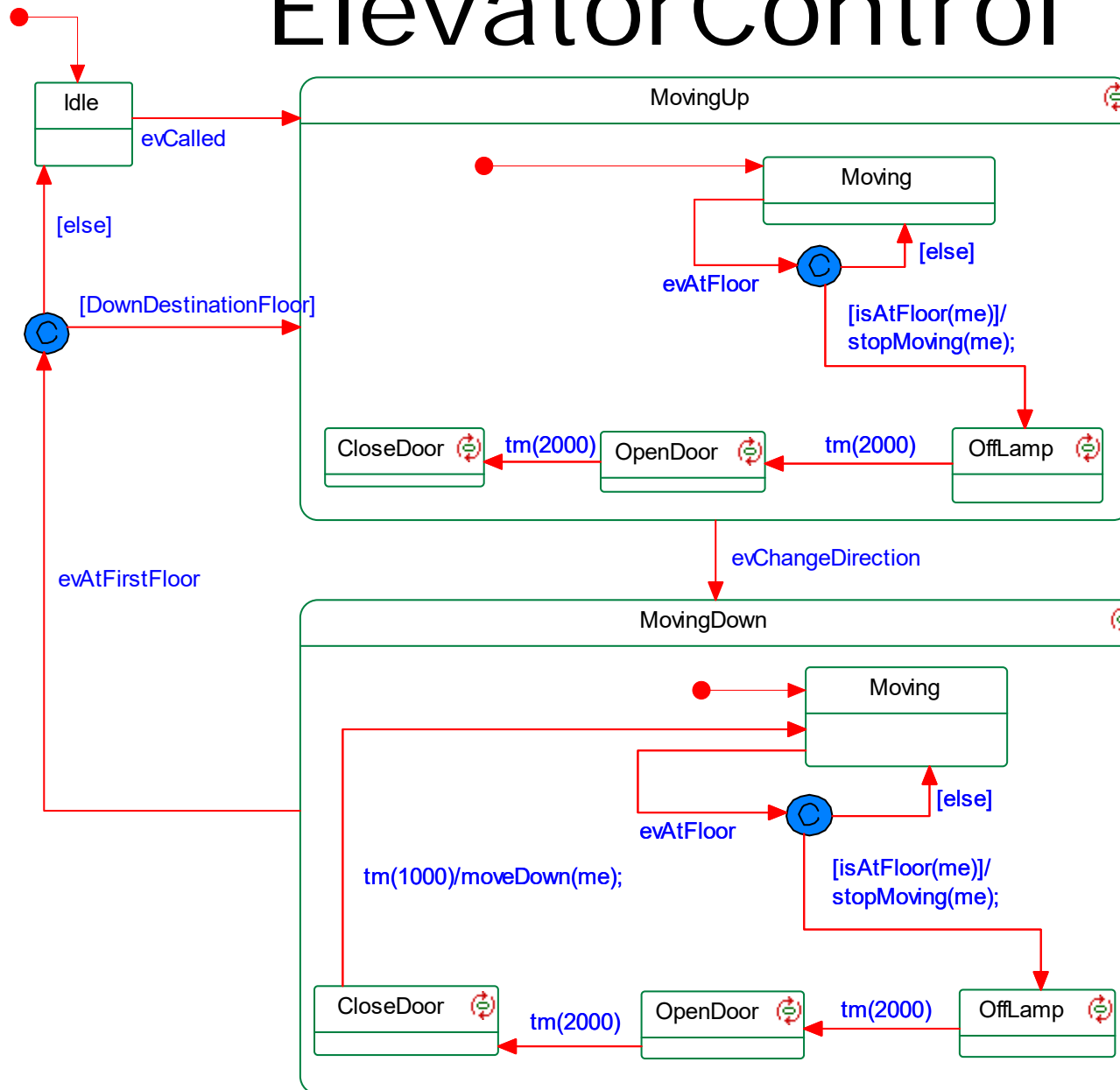
# ReceiveMessageObj



# Motor



# ElevatorControl

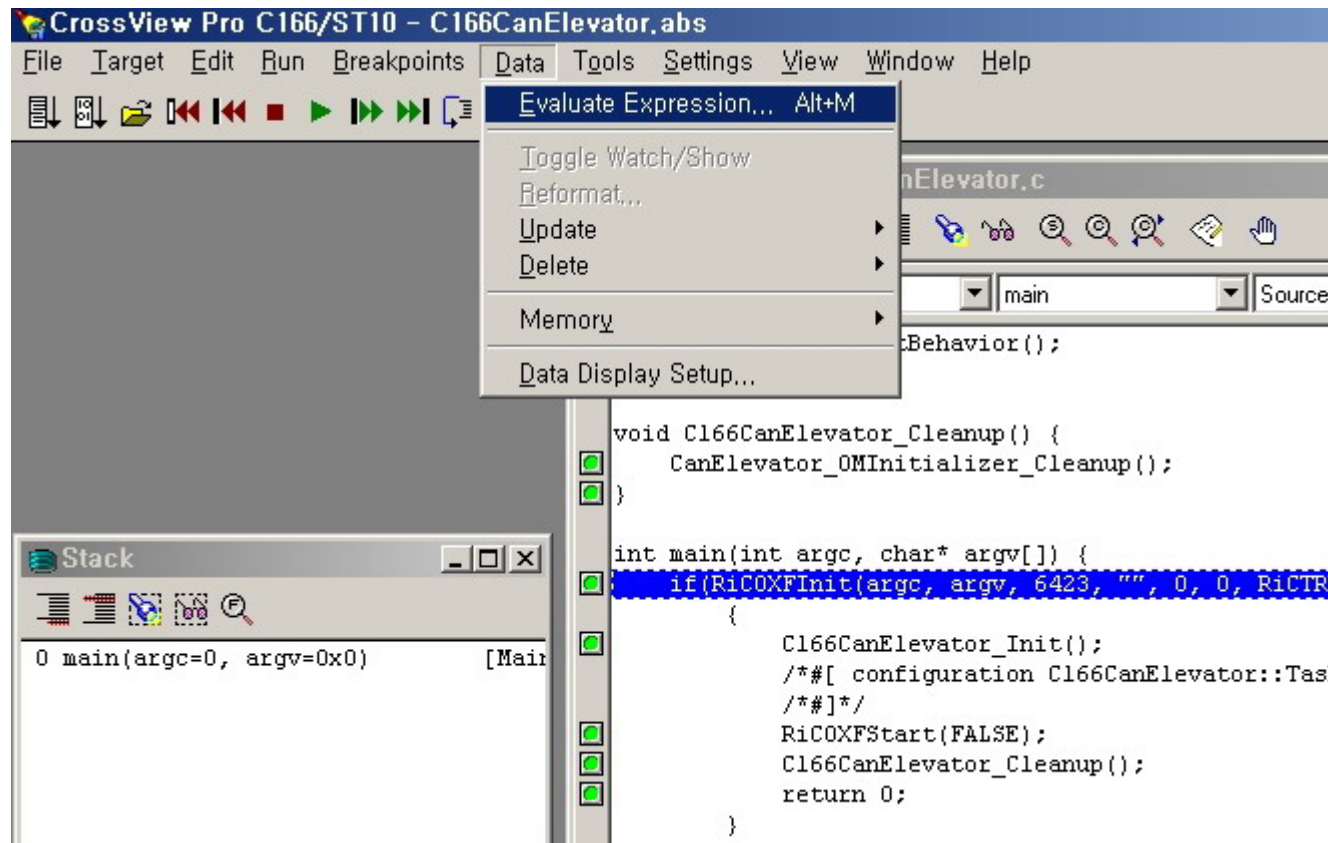




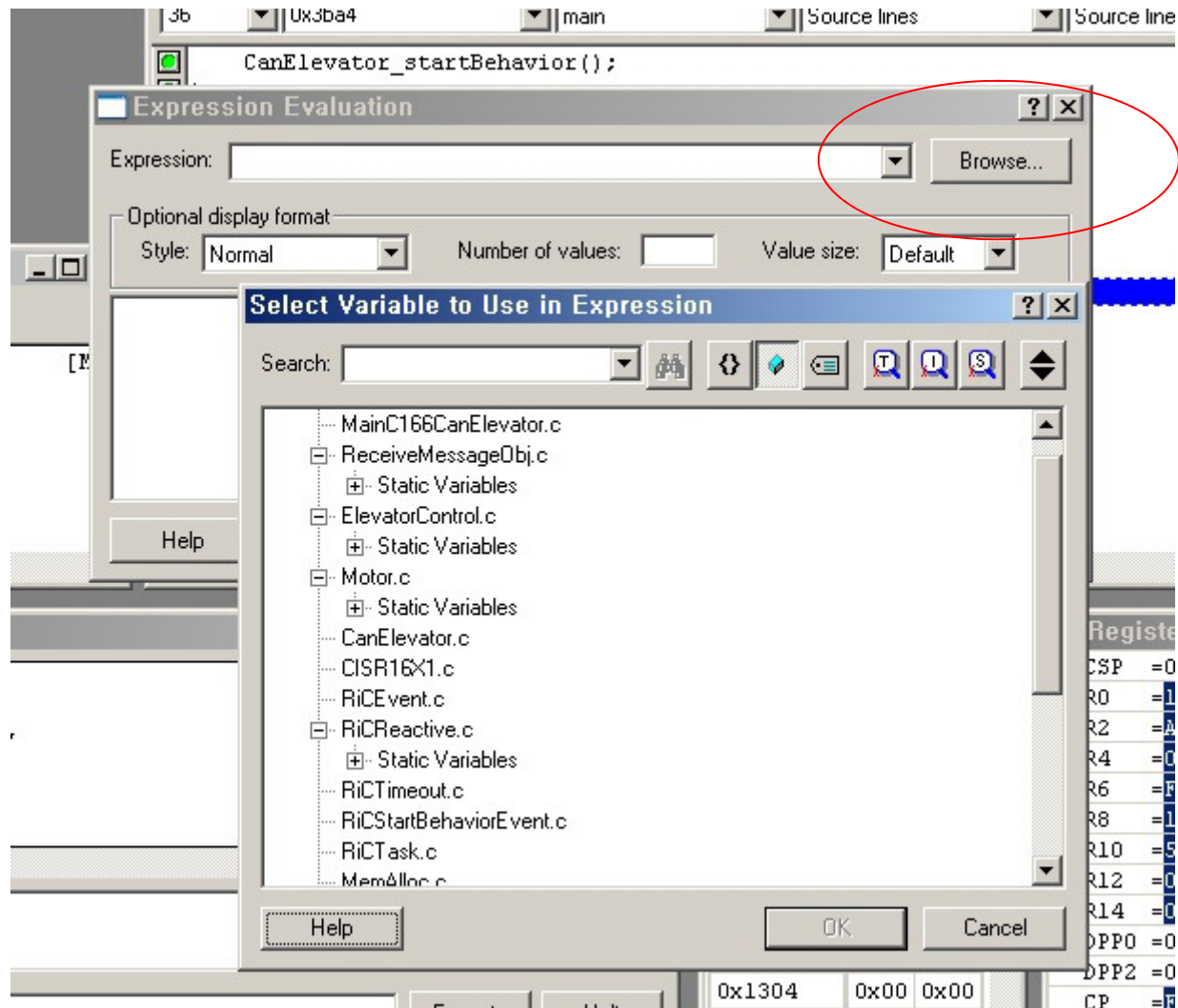
# Caution

- When CAN messages are received while the program is not ready to accept messages, an error occurs. In that case, start over the program again.
- Be cautious when using *printf*, since it slows down the program and misses CAN messages.

# CrossView Debugger



# CrossView Debugger



# CrossView Debugger

