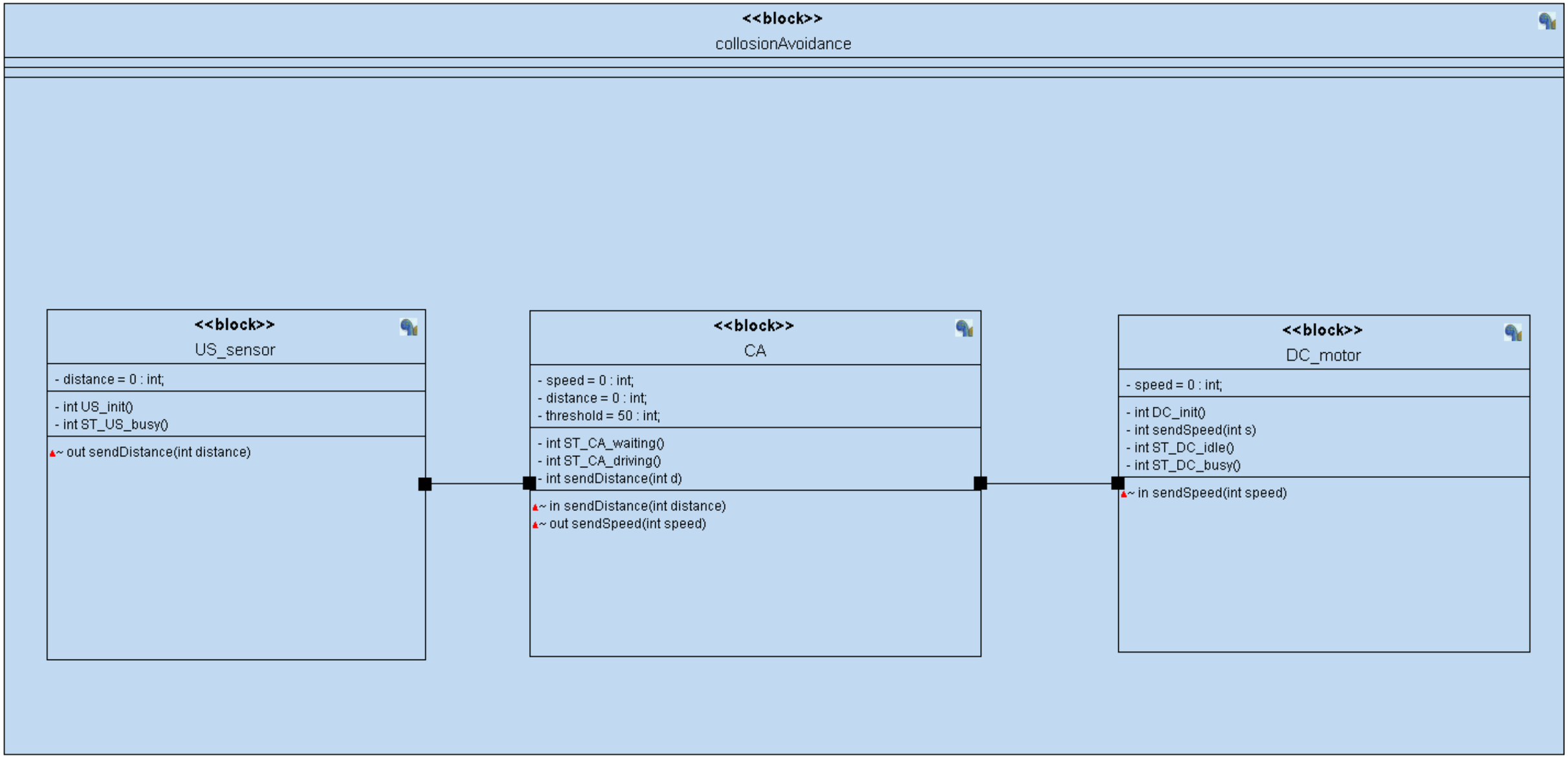
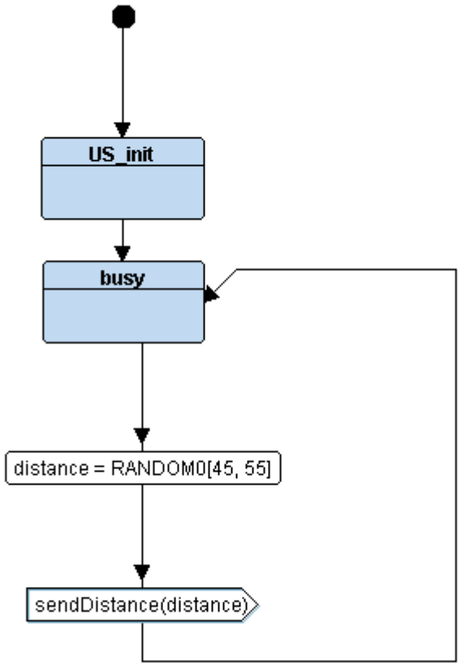
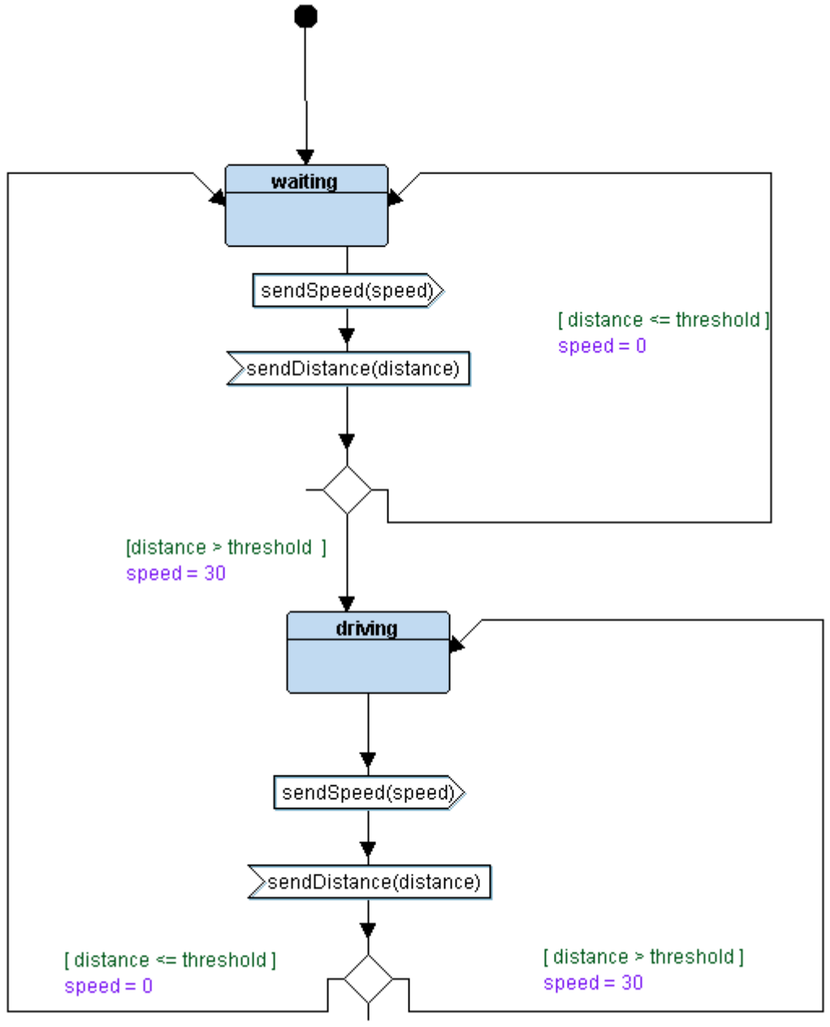
MODULES

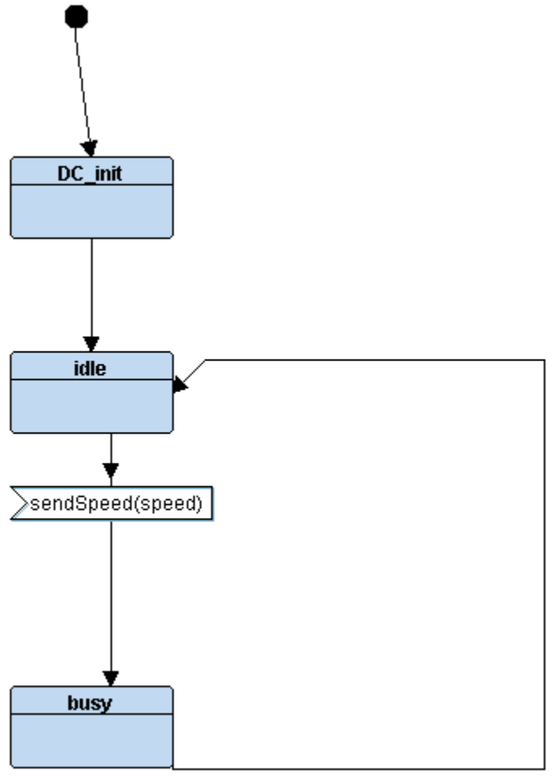

<<block>>
collosionAvoidance
<<block>>
US_sensor
- distance = 0 : int;
- int US_init()
- int ST_US_busy()
~ out sendDistance(int distance)
<<block>>
CA
- speed = 0 : int;
- distance = 0 : int;
- threshold = 50 : int;
- int ST_CA_waiting()
- int ST_CA_driving()
- int sendDistance(int d)
~ in sendDistance(int distance)
~ out sendSpeed(int speed)
<<block>>
DC_motor
- speed = 0 : int;
- int DC_init()
- int sendSpeed(int s)
- int ST_DC_idle()
- int ST_DC_busy()
~ in sendSpeed(int speed)

`

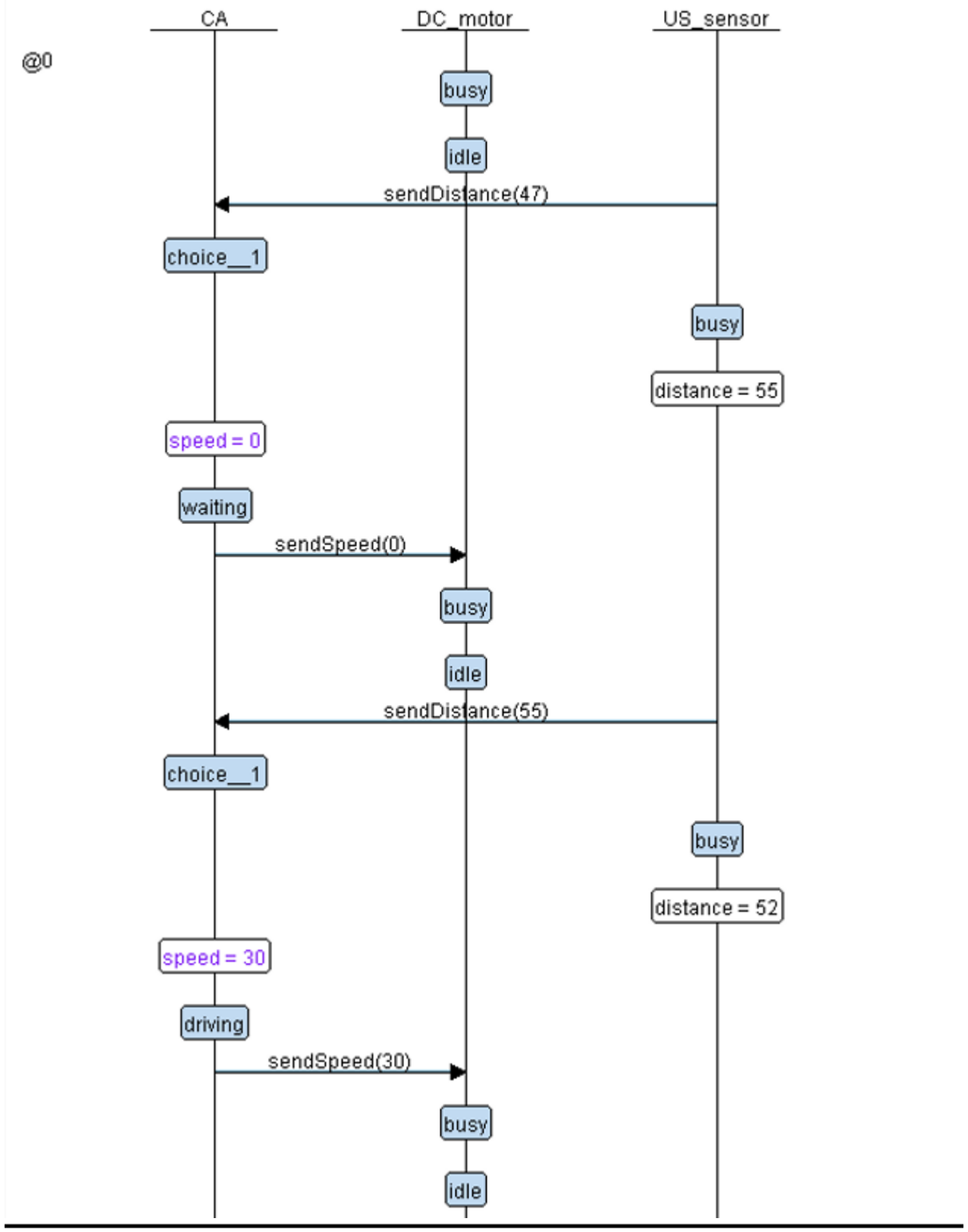
DIAGRAMS


US_init
busy
distance = RANDOM0[45, 55]
sendDistance(distance)


waiting
sendSpeed(speed)
sendDistance(distance)
[distance <= threshold]
speed = 0
[distance > threshold]
speed = 30
driving
sendSpeed(speed)
sendDistance(distance)
[distance <= threshold]
speed = 0
[distance > threshold]
speed = 30


DC_init
idle
sendSpeed(speed)
busy

Logical Verification



`

```
US_init
DC_init
US ----- distance = 42 ------> CA
CA ----- Speed = 0 ------> DC
CA_waiting State distance = 42 , Speed = 0
DC_busy state :  Speed = 0
US ----- distance = 43 ------> CA
CA ----- Speed = 0 ------> DC
CA_waiting State distance = 43 , Speed = 0
DC_busy state :  Speed = 0
US ----- distance = 10 ------> CA
CA ----- Speed = 0 ------> DC
CA_waiting State distance = 10 , Speed = 0
DC_busy state :  Speed = 0
US ----- distance = 46 ------> CA
CA ----- Speed = 0 ------> DC
CA_waiting State distance = 46 , Speed = 0
DC_busy state :  Speed = 0
US ----- distance = 30 ------> CA
CA ----- Speed = 0 ------> DC
CA_waiting State distance = 30 , Speed = 0
DC_busy state :  Speed = 0
US ----- distance = 50 ------> CA
CA ----- Speed = 0 ------> DC
CA_waiting State distance = 50 , Speed = 0
DC_busy state :  Speed = 0
```

`