Tala Nemeh

101170694

Department of Systems and Computer Engineering

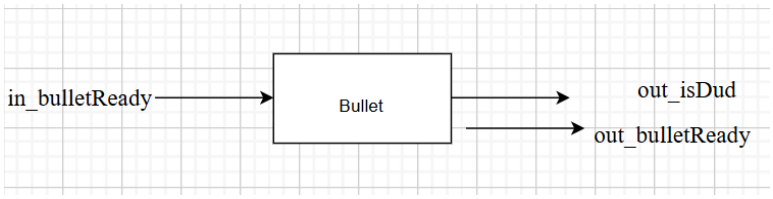
Course: SYSC4906

Assignment 1

Introduction:
This report presents the simulation and analysis of a discrete-event model developed using the DEVS (Discrete Event System Specification) formalism. The main objective of this assignment is to design, implement, and evaluate a simulation model that accurately represents the behavior of a system using the Cadmium DEVS simulation framework.

Devs Models Specifications:

Bullet (Atomic Model):



$M_{Bullet} = \langle X, S, Y, \delta_{ext}, \delta_{int}, \lambda, ta \rangle$

X: { in_bulletReady }

S: { currentState ∈ { PASSIVE, ACTIVE }, bulletRdy ∈ {0, 1}, isDud ∈ {0, 1}

Y: { out_isDud, out_bulletReady }

δ_{ext}: Processes the message from in_bulletReady and determines the dud status randomly (95% chance of a good bullet). It then schedules an immediate internal transition ($\sigma = 0$).

Pseudocode:

```
if (in_bulletReady not empty) then
    bulletRdy = last message from in_bulletReady

randVal = random value in [0,1]
if (randVal < 0.95) then
    isDud = 0   // bullet is good
else
    isDud = 1   // bullet is a dud

currentState = ACTIVE
σ = 0  // schedule immediate internal transition
```

δ_{int}: Resets the model to a passive state by setting currentState to PASSIVE and σ to ∞

```
currentState = PASSIVE
σ = ∞
```

λ: Outputs the dud status and the bullet-ready flag based on the current state.

Pseudocode:

```
send out_isDud = isDud
send out_bulletReady = bulletRdy
```

ta: Returns σ

Chamber (Atomic model):

$M_{Chamber} = \langle X, S, Y, \delta_{ext}, \delta_{int}, \lambda, ta \rangle$

X: { in_isDud, in_bulletLoaded}

S: {currentState ∈ { passive, active },dudBullet, bulletIn}

Y:{out_boltBack, out_bulletFired, out_casing}

δ_{ext}: Processes incoming messages as follows:

Pseudocode:

```
if in_isDud has a message:
    dudBullet = message

if in_bulletLoaded has a message:
    bulletIn = message
    sigma = 5.0
```

δ_{int}: On an internal event, the model resets:

Pseudocode:

```
currentState = passive
dudBullet = 2      // clear the dudBullet variable
bulletIn = 0       // clear the bulletIn variable
sigma = infinity   // no further events scheduled
```

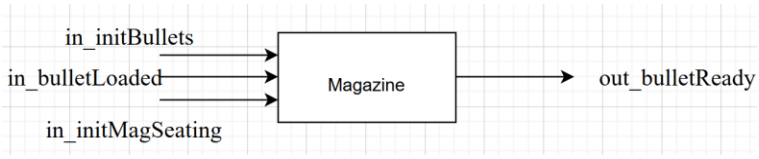
λ:

Pseudocode:

```
if dudBullet == 0 and bulletIn == 1:
    send out_boltBack message 1
    send out_bulletFired message 1
    send out_casing message 1
```

ta: Returns σ

Magazine(Atomic model):



$M_{Magazine} = \langle X, S, Y, \delta_{ext}, \delta_{int}, \lambda, ta \rangle$

X: { in_initBullets, in_bulletLoaded, in_initMagSeating }

S: {currentState ∈ { passive, active }, tempMsgVal, bulletsLeft, magSeating, bulletReady}

Y: { out_bulletReady }

δ_{ext}:

```
sigma = sigma - e

if in_initBullets has a message:
    tempMsgVal = message
    if tempMsgVal >= 0 and tempMsgVal < 30:
        bulletsLeft = tempMsgVal

else if in_initMagSeating has a message:
    tempMsgVal = message
    magSeating = tempMsgVal

else if in_bulletLoaded has a message:
    tempMsgVal = message
    if tempMsgVal == 1:
        bulletsLeft = bulletsLeft - 1

if bulletsLeft >= 0:
    if magSeating == 1:
        bulletReady = 1
else:
    bulletReady = 0

currentState = active
sigma = 0.0
```

δ_{int}:

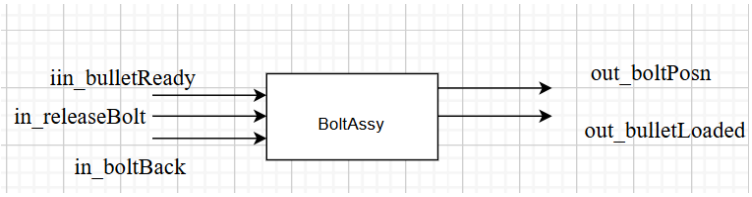
```
currentState = passive
sigma = infinity
```

λ:

```
send out_bulletReady message = bulletReady
```

ta: Returns σ

BoltAssy:



$M_{BoltAssy}$ = <X,S,Y, δ_{ext}, δ_{int}, λ, ta>

X: { in_bulletReady, in_releaseBolt, in_boltBack } (messages of type int)

S: { currentState ∈ { PASSIVE, ACTIVE }, boltFree ∈ {0,1}, readyBullet ∈ {0,1}, boltState ∈ {0 (forward), 1 (back), 2 (error)} }

Y: { out_bulletLoaded, out_boltPosn }

δ_{ext}: Processes messages from in_bulletReady, in_releaseBolt, and in_boltBack; schedules an immediate transition (σ = 0) after updating boltState.

Pseudocode:

```
if in_bulletReady has a message:
    readyBullet = message

if in_releaseBolt has a message && message == 1 && boltState == 1:
    boltFree = 1

if in_boltBack has a message && message == 1 :
    if boltState equals 0 :
        boltState = 1
    else if boltState == 1:
        boltState = 0

if boltFree == 1 && boltState == 1 :
    randVal = random value between 0 and 1
    if readyBullet == 1
        if randVal < 0.90:
            boltState = 0   // successful load
        else
            boltState = 2   // loading issue
    else
        boltState = 0       // no bullet ready

    readyBullet = 0
    boltFree = 0
    currentState = active
    sigma = 0.0
```

δ_{int}: Resets the model to a passive state (currentState = PASSIVE, σ = ∞).

Pseudocode:

```
currentState = PASSIVE
σ = ∞
```

λ: Outputs bullet load status and bolt position based on the current boltState.

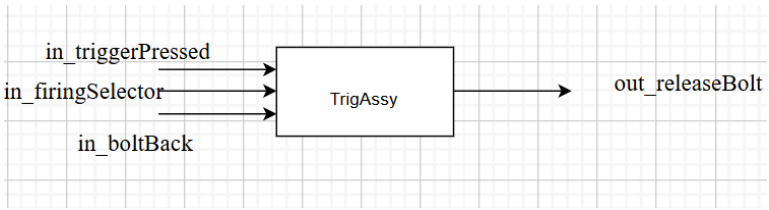
Pseudocode:

```
if (state.boltState == 0) then
    send out_bulletLoaded = 1
else if (state.boltState == 2) then
    send out_bulletLoaded = 0

send out_boltPosn = state.boltState
```

ta: Returns σ.

TrigAssy(Atomic model):



TrigAssy = < X, S, Y, δ_ext, δ_int, λ, ta >

X:{ in_triggerPressed, in_firingSelector, in_boltBack}

S: {currentState ∈ { passive, active }, triggerPull ∈ { 0, 1 }, firingSelector ∈ { 0, 1, 2 }}

Y: { out_releaseBolt}

δ_ext:

```
if in_triggerPressed has a message then
        triggerPull = last message from in_triggerPressed
    if in_firingSelector has a message then
        firingSelector = last message from in_firingSelector
    if in_boltBack has a message then
        // no direct state change; used to schedule output event
    currentState = active
    sigma = 0.0
```

δ_int: Resets the model to a passive state by setting currentState to PASSIVE and σ to ∞

```
if firingSelector equals 1 and triggerPull equals 1 then
        triggerPull = 0
    currentState = passive
    sigma = infinity
```

λ:

```
if triggerPull does not equal 1 then
        do not produce output
    else
        if firingSelector equals 0 then
            send out_releaseBolt message 0
        else if firingSelector equals 1 then
            send out_releaseBolt message 1
        else if firingSelector equals 2 then
            send out_releaseBolt message 1
```

ta: Returns σ.

MaggAssy(Coupled model):

MagAssy = ⟨X, Y, {M_i}, EIC, EOC, IC⟩

X:{in_initBullets, in_initMagSeating, in_bulletLoaded}

Y:{ out_bulletReady,out_isDud}

Submodels ({M_i}):

Magazine (an atomic DEVS model, as defined in Magazine.hpp)

Bullet (an atomic DEVS model, as defined in Bullet.hpp)

EIC: (

in_initBullets → Magazine.in_initBullets

in_initMagSeating → Magazine.in_initMagSeating

in_bulletLoaded → Magazine.in_bulletLoaded)

IC:

(Magazine.out_bulletReady → Bullet.in_bulletReady)

EOC:

(Bullet.out_bulletReady → out_bulletReady

Bullet.out_isDud → out_isDud)

Rifle (Coupled model):

Rifle = ⟨X, Y, {M_i}, EIC, EOC, IC⟩

X: { in_triggerPressed, in_firingSelector, in_boltBack, in_magSeating, in_bulletLoaded }

Y: { out_releaseBolt }

Submodels ({M_i}):

MagAssy (a coupled DEVS model, as defined in MagAssy.hpp)

TrigAssy (an atomic DEVS model, as defined in TrigAssy.hpp)

BoltAssy (an atomic DEVS model, as defined in BoltAssy.hpp)

Chamber (an atomic DEVS model, as defined in Chamber.hpp)

EIC: (

in_triggerPressed → TrigAssy.in_triggerPressed

in_firingSelector → TrigAssy.in_firingSelector

in_boltBack → BoltAssy.in_boltBack

in_magSeating → MagAssy.in_initMagSeating

in_bulletLoaded → MagAssy.in_bulletLoaded)

IC: (

MagAssy.out_bulletReady → BoltAssy.in_bulletReady

MagAssy.out_isDud → Chamber.in_isDud

TrigAssy.out_releaseBolt → BoltAssy.in_releaseBolt

BoltAssy.out_bulletLoaded → Chamber.in_bulletLoaded

BoltAssy.out_bulletLoaded → MagAssy.in_bulletLoaded

Chamber.out_boltBack → BoltAssy.in_boltBack

Chamber.out_boltBack → TrigAssy.in_boltBack)

EOC: (TrigAssy.out_releaseBolt → out_releaseBolt)

Test Scenarios and Results

The testing phase was designed to evaluate the accuracy and efficiency of the developed DEVS models. The test scenarios were designed to assess the behavior of the system under different conditions, ensuring correct state transitions and input-output consistency. Each scenario was designed to validate individual atomic models before conducting system-wide integration tests.

Atomic models tests:

Bullet Model:

Bullet Queue Test

Objective: To verify how the Bullet model handles a queue of bullets being ready sequentially without time gaps between signals, testing its capacity to handle rapid transitions.

Method:

- **Setup:** The generator will send a rapid sequence of bullet ready signals (in_bulletReady = 1) in quick succession.
- **Expected Behavior:** The Bullet model should process each input without error, managing state transitions between ACTIVE and PASSIVE effectively.

Execution Steps:

- Send multiple (bulletReady = 1) signals in rapid succession.

```
time;model_id;model_name;port_name;data
0;1;bulletmodel;;{1, State: PASSIVE, bulletRdy: 0, isDud: 0}
0;2;genmodel;;{Messages Sent: 0, Sigma: 1}
[BulletQueueGenerator] Sending bulletReady: 1 (message #1)
1;1;bulletmodel;out_isDud;0
1;1;bulletmodel;out_bulletReady;0
1;1;bulletmodel;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
1;2;genmodel;out_bulletReady;1
1;2;genmodel;;{Messages Sent: 1, Sigma: 1}
1;1;bulletmodel;out_isDud;0
1;1;bulletmodel;out_bulletReady;1
1;1;bulletmodel;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
[BulletQueueGenerator] Sending bulletReady: 1 (message #2)
2;1;bulletmodel;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
2;2;genmodel;out_bulletReady;1
2;2;genmodel;;{Messages Sent: 2, Sigma: 1}
2;1;bulletmodel;out_isDud;0
2;1;bulletmodel;out_bulletReady;1
2;1;bulletmodel;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
[BulletQueueGenerator] Sending bulletReady: 1 (message #3)
3;1;bulletmodel;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
3;2;genmodel;out_bulletReady;1
3;2;genmodel;;{Messages Sent: 3, Sigma: 1}
3;1;bulletmodel;out_isDud;0
```

The Bullet model correctly responds to bullet readiness signals, processes inputs, and transitions between states as expected.

State Transitions:

The Bullet model starts in a PASSIVE state (bulletRdy = 0, isDud = 0).

Upon receiving a bulletReady = 1 message, it transitions to ACTIVE (bulletRdy = 1).

After processing the input, the model outputs out_bulletReady = 1 and out_isDud = 0, then returns to the PASSIVE state.

Bullet Readiness and Output:

Each bulletReady message is processed sequentially, and the model outputs a valid response.

The transitions occur without delay, confirming that the model correctly processes input signals and updates states.

Repeated Execution:

The log shows a consistent cycle where the model transitions from PASSIVE → ACTIVE → PASSIVE for each input signal.

This confirms that the Bullet model handles repeated inputs as expected.

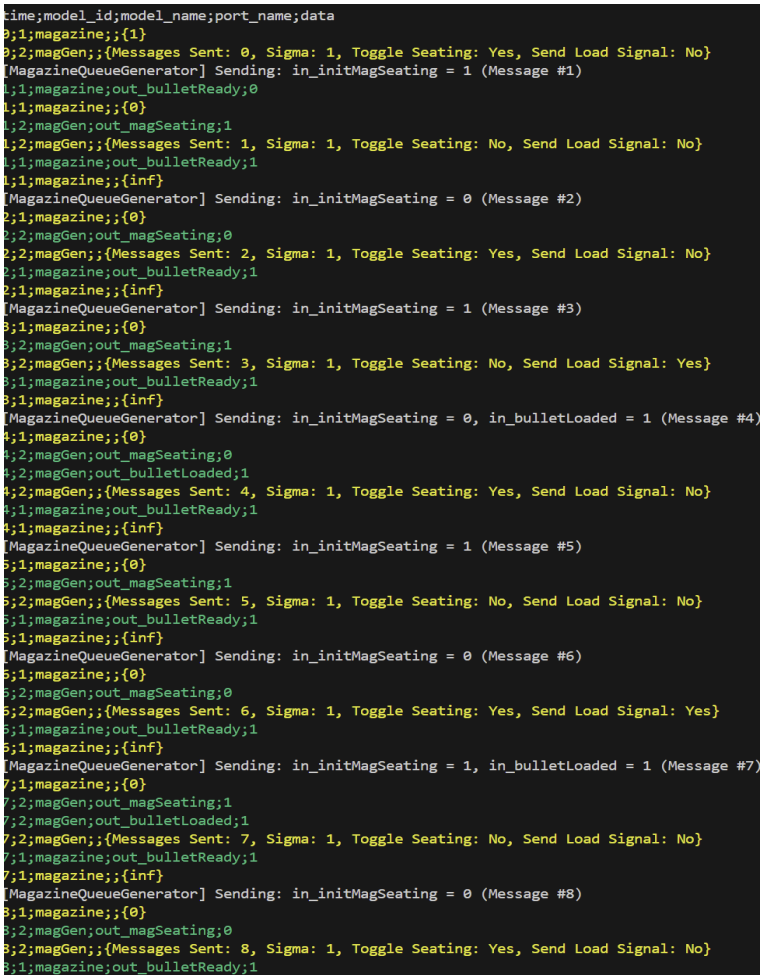
Magazine Model:

Magazine Rapid Re-Seating Test

Objective: To verify the Magazine model's ability to handle frequent changes in seating status while correctly tracking bullet readiness and count.

Method:

- **Setup**: The generator rapidly toggles in_initMagSeating between seated (1) and unseated (0), while occasionally sending in_bulletLoaded = 1 signals.
- **Expected Behavior**:
    - When seated and bullets remain, out_bulletReady should be 1.
    - If unseated or out of bullets, out_bulletReady should be 0.
    - Bullet count decrements only when in_bulletLoaded = 1 arrives while seated.
- **Execution Steps:**
    - Rapidly switch in_initMagSeating between 1 and 0.
    - Send in_bulletLoaded = 1 at intermittent intervals.
    - Observe out_bulletReady after each seating toggle and loading signal to confirm correct state transitions and bullet count management.
- **Results**:



When seated, the Magazine correctly sets out_bulletReady = 1.

When unseated, out_bulletReady = 0, preventing bullets from being loaded.

Bullet load signals only affect the bullet count when the magazine is seated.

The Magazine model successfully responds to rapid seating changes while ensuring bullets are only loaded when seated, demonstrating correct state transitions and bullet management under stress conditions.

Bolt Assembly

Bolt Cycling and Release Test

Objective: To verify how the Bolt Assembly (BoltAssy) model handles rapid bolt movements, ensuring correct state transitions and bullet loading behavior under frequent activation.

Method:

- **Setup**: The generator rapidly toggles the bolt position (in_boltBack) between pulled back (1) and forward (0), while occasionally sending in_releaseBolt = 1 to simulate bolt release.
- **Expected Behavior**:
    - When bolt is back (1) and a bullet is ready, out_bulletLoaded should be 1.
    - When bolt is forward (0), out_bulletLoaded should be 0, preventing bullet loading.
    - The model should correctly track the bolt state and respond accurately to in_releaseBolt.
- **Execution Steps**
    - Toggle in_boltBack between 1 and 0 to simulate rapid bolt movement.
    - Send in_releaseBolt = 1 intermittently to test proper interaction between bolt release and bullet loading.
- **Results:**

```
time;model_id;model_name;port_name;data
0;1;boltAssy;;{1, State: ACTIVE, boltFree: 0, readyBullet: 0, boltState: 0}
0;2;boltGen;;{Messages Sent: 0, Sigma: 1, Toggle Bolt: Yes, Send Release Signal: No}
[BoltAssyQueueGenerator] Sending: in_boltBack = 1 (Message #1)
1;1;boltAssy;out_bulletLoaded;1
1;1;boltAssy;out_boltPosn;0
1;1;boltAssy;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 0, boltState: 1}
1;2;boltGen;out_boltBack;1
1;2;boltGen;;{Messages Sent: 1, Sigma: 1, Toggle Bolt: No, Send Release Signal: No}
[BoltAssyQueueGenerator] Sending: in_boltBack = 0 (Message #2)
2;1;boltAssy;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 0, boltState: 1}
2;2;boltGen;out_boltBack;0
2;2;boltGen;;{Messages Sent: 2, Sigma: 1, Toggle Bolt: Yes, Send Release Signal: No}
[BoltAssyQueueGenerator] Sending: in_boltBack = 1 (Message #3)
3;1;boltAssy;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 0, boltState: 0}
3;2;boltGen;out_boltBack;1
3;2;boltGen;;{Messages Sent: 3, Sigma: 1, Toggle Bolt: No, Send Release Signal: No}
[BoltAssyQueueGenerator] Sending: in_boltBack = 0 (Message #4)
4;1;boltAssy;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 0, boltState: 0}
4;2;boltGen;out_boltBack;0
4;2;boltGen;;{Messages Sent: 4, Sigma: 1, Toggle Bolt: Yes, Send Release Signal: Yes}
[BoltAssyQueueGenerator] Sending: in_boltBack = 1, in_releaseBolt = 1 (Message #5)
5;1;boltAssy;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 0, boltState: 0}
5;2;boltGen;out_boltBack;1
5;2;boltGen;out_releaseBolt;1
5;2;boltGen;;{Messages Sent: 5, Sigma: 1, Toggle Bolt: No, Send Release Signal: No}
[BoltAssyQueueGenerator] Sending: in_boltBack = 0 (Message #6)
6;1;boltAssy;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 0, boltState: 0}
6;2;boltGen;out_boltBack;0
6;2;boltGen;;{Messages Sent: 6, Sigma: 1, Toggle Bolt: Yes, Send Release Signal: No}
[BoltAssyQueueGenerator] Sending: in_boltBack = 1 (Message #7)
7;1;boltAssy;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 0, boltState: 1}
7;2;boltGen;out_boltBack;1
7;2;boltGen;;{Messages Sent: 7, Sigma: 1, Toggle Bolt: No, Send Release Signal: No}
[BoltAssyQueueGenerator] Sending: in_boltBack = 0 (Message #8)
8;1;boltAssy;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 0, boltState: 1}
8;2;boltGen;out_boltBack;0
8;2;boltGen;;{Messages Sent: 8, Sigma: 1, Toggle Bolt: Yes, Send Release Signal: Yes}
[BoltAssyQueueGenerator] Sending: in_boltBack = 1, in_releaseBolt = 1 (Message #9)
```

The model correctly updates out_boltPosn based on in_boltBack.
out_bulletLoaded is only triggered when the bolt is back and a bullet is available.
out_releaseBolt is correctly triggered when in_releaseBolt = 1.

The BoltAssy model successfully tracks rapid bolt movements while ensuring bullets are only loaded under the correct conditions. The test confirms that frequent state transitions do not cause errors or inconsistencies.

Trigger Assembly

Trigger Mode and Rapid Fire Test

Objective: To verify how the Trigger Assembly (TrigAssy) model handles rapid trigger pulls (in_triggerPressed = 1) and firing mode changes (in_firingSelector), ensuring proper bolt release behavior.

Method:

- **Setup**: The generator will rapidly toggle the trigger state (in_triggerPressed = 1 or 0) while intermittently changing the firing mode (in_firingSelector = 0, 1, or 2).
- **Expected Behavior**:
  - If the trigger is pulled (in_triggerPressed = 1) in SAFE mode (0), no bolt release occurs.
  - In SINGLE mode (1), one bolt release occurs per trigger pull.
  - In AUTO mode (2), the trigger remains engaged, allowing continuous bolt releases.
  - The model should properly transition between states without missing trigger inputs.
- **Execution Steps:**
  - Toggle in_triggerPressed between 1 and 0 to simulate rapid trigger pulls.
  - Switch in_firingSelector between 0 (safe), 1 (single), and 2 (auto) at various intervals.

- **Results:**

```
time;model_id;model_name;port_name;data
0;1;trigAssy;;{inf, 0}
0;2;trigGen;;{Messages Sent: 0, Sigma: 1, Toggle Trigger: Yes, Firing Mode: 1}
[TriggerQueueGenerator] Sending: in_triggerPressed = 1, in_firingSelector = 1 (Message #1)
1;1;trigAssy;;{0, 1}
1;2;trigGen;out_triggerPressed;1
1;2;trigGen;out_firingSelector;1
1;2;trigGen;;{Messages Sent: 1, Sigma: 1, Toggle Trigger: No, Firing Mode: 1}
1;1;trigAssy;out_releaseBolt;1
1;1;trigAssy;;{inf, 0}
[TriggerQueueGenerator] Sending: in_triggerPressed = 0, in_firingSelector = 1 (Message #2)
2;1;trigAssy;;{0, 1}
2;2;trigGen;out_triggerPressed;0
2;2;trigGen;out_firingSelector;1
2;2;trigGen;;{Messages Sent: 2, Sigma: 1, Toggle Trigger: Yes, Firing Mode: 1}
2;1;trigAssy;;{inf, 0}
[TriggerQueueGenerator] Sending: in_triggerPressed = 1, in_firingSelector = 1 (Message #3)
3;1;trigAssy;;{0, 1}
3;2;trigGen;out_triggerPressed;1
3;2;trigGen;out_firingSelector;1
3;2;trigGen;;{Messages Sent: 3, Sigma: 1, Toggle Trigger: No, Firing Mode: 2}
3;1;trigAssy;out_releaseBolt;1
3;1;trigAssy;;{inf, 0}
[TriggerQueueGenerator] Sending: in_triggerPressed = 0, in_firingSelector = 2 (Message #4)
4;1;trigAssy;;{0, 1}
4;2;trigGen;out_triggerPressed;0
4;2;trigGen;out_firingSelector;2
4;2;trigGen;;{Messages Sent: 4, Sigma: 1, Toggle Trigger: Yes, Firing Mode: 2}
4;1;trigAssy;;{inf, 0}
[TriggerQueueGenerator] Sending: in_triggerPressed = 1, in_firingSelector = 2 (Message #5)
5;1;trigAssy;;{0, 1}
5;2;trigGen;out_triggerPressed;1
5;2;trigGen;out_firingSelector;2
5;2;trigGen;;{Messages Sent: 5, Sigma: 1, Toggle Trigger: No, Firing Mode: 2}
5;1;trigAssy;out_releaseBolt;1
5;1;trigAssy;;{inf, 0}
[TriggerQueueGenerator] Sending: in_triggerPressed = 0, in_firingSelector = 2 (Message #6)
6;1;trigAssy;;{0, 1}
6;2;trigGen;out_triggerPressed;0
6;2;trigGen;out_firingSelector;2
6;2;trigGen;;{Messages Sent: 6, Sigma: 1, Toggle Trigger: Yes, Firing Mode: 0}
6;1;trigAssy;;{inf, 0}
[TriggerQueueGenerator] Sending: in_triggerPressed = 1, in_firingSelector = 0 (Message #7)
7;1;trigAssy;;{0, 1}
7;2;trigGen;out_triggerPressed;1
7;2;trigGen;out_firingSelector;0
7;2;trigGen;;{Messages Sent: 7, Sigma: 1, Toggle Trigger: No, Firing Mode: 0}
```

SAFE mode (0) correctly prevents out_releaseBolt.
SINGLE mode (1) allows one bolt release per trigger pull.
AUTO mode (2) correctly maintains firing when the trigger remains pressed.

The TrigAssy model correctly responds to rapid mode changes, ensuring that the bolt is released only under valid conditions.

Coupled Model

Magazine Assembly Loading and Readiness Test

Objective

To verify how the MagAssy model handles bullet loading (in_bulletLoaded), magazine seating (in_initMagSeating), and bullet readiness (out_bulletReady), ensuring correct operation when interacting with the Bullet model.

Method

- **Setup**:
  - The generator alternates between seated (1) and unseated (0) magazine states.
  - Sends bullet load requests (in_bulletLoaded = 1) at different intervals.
  - Starts with a set number of bullets (in_initBullets) to test bullet depletion.
- **Expected Behavior**:
  - If magazine is seated and bullets remain, out_bulletReady = 1.
  - If magazine is unseated, out_bulletReady = 0, even if bullets remain.
  - Each bullet load request (in_bulletLoaded = 1) decreases the count and updates out_bulletReady.
  - Once bullets reach zero, out_bulletReady should remain 0, preventing further bullet usage.
- **Execution Steps**
  - Initialize the system:
    Set in_initBullets = 10 (start with 10 bullets) and in_initMagSeating = 1 (seated).
  - Toggle magazine seating:
    Send seating/unseating signals (1 → 0 → 1) to check if out_bulletReady updates correctly.
  - Load bullets:
  - Send in_bulletLoaded = 1, ensuring bullet count decreases properly.
- **Results:**

out_bulletReady = 1 when the magazine is seated and bullets are available.
out_bulletReady = 0 when the magazine is unseated, even if bullets remain.
Bullet count decreases correctly with in_bulletLoaded = 1, ensuring proper tracking.

The MagAssy model correctly updates bullet availability based on magazine seating and bullet loading, preventing incorrect firings when unseated or empty.

System Test

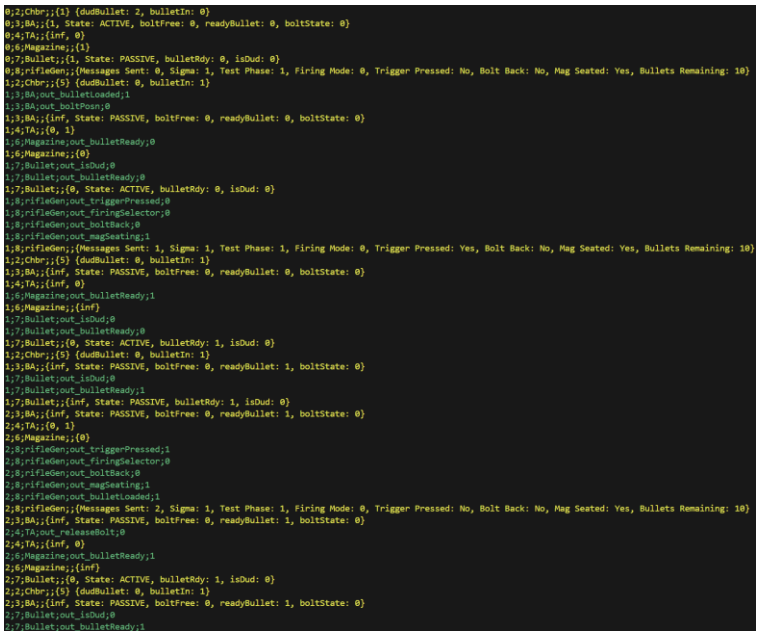
Test 1: Firing Mode and Trigger Test

Objective: Test three scenarios to determine whether the full system functions correctly under different test cases:

Scenario 1 tests the basic trigger and firing mode change.

Scenario 2 examines how the system reacts to changes in magazine seating and bullet availability.

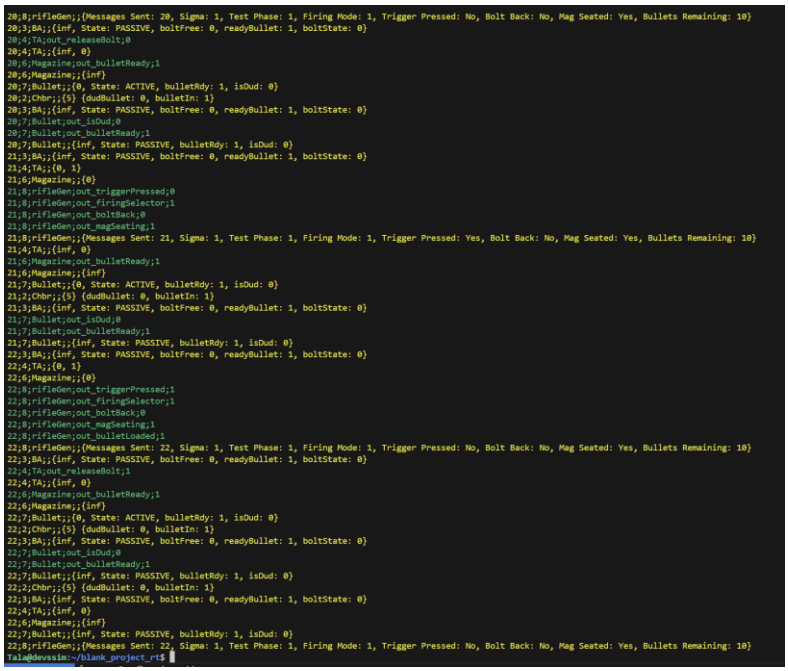
Scenario 3 tests the effect of bolt position toggling on the firing process.



```
4;8;rifleGen;out_triggerPressed;1
4;8;rifleGen;out_firingSelector;0
4;8;rifleGen;out_boltBack;0
4;8;rifleGen;out_magSeating;1
4;8;rifleGen;out_bulletLoaded;1
4;8;rifleGen;;{Messages Sent: 4, Sigma: 1, Test Phase: 1, Firing Mode: 0, Trigger Pressed: No, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
4;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
4;4;TA;out_releaseBolt;0
4;4;TA;;{inf, 0}
4;6;Magazine;out_bulletReady;1
4;6;Magazine;;{inf}
4;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
4;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
4;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
4;7;Bullet;out_isDud;0
4;7;Bullet;out_bulletReady;1
4;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
5;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
5;4;TA;;{0, 1}
5;6;Magazine;;{0}
5;8;rifleGen;out_triggerPressed;0
5;8;rifleGen;out_firingSelector;0
5;8;rifleGen;out_boltBack;0
5;8;rifleGen;out_magSeating;1
5;8;rifleGen;;{Messages Sent: 5, Sigma: 1, Test Phase: 1, Firing Mode: 1, Trigger Pressed: Yes, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
5;4;TA;;{inf, 0}
5;6;Magazine;out_bulletReady;1
5;6;Magazine;;{inf}
5;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
5;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
5;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
5;7;Bullet;out_isDud;0
5;7;Bullet;out_bulletReady;1
5;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
6;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
6;4;TA;;{0, 1}
6;6;Magazine;;{0}
6;8;rifleGen;out_triggerPressed;1
6;8;rifleGen;out_firingSelector;1
6;8;rifleGen;out_boltBack;0
6;8;rifleGen;out_magSeating;1
6;8;rifleGen;out_bulletLoaded;1
6;8;rifleGen;;{Messages Sent: 6, Sigma: 1, Test Phase: 1, Firing Mode: 1, Trigger Pressed: No, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
6;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
6;4;TA;out_releaseBolt;1
6;4;TA;;{inf, 0}
6;6;Magazine;out_bulletReady;1
6;6;Magazine;;{inf}
6;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
6;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
6;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
6;7;Bullet;out_isDud;0
6;7;Bullet;out_bulletReady;1
6;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
7;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
7;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
7;4;TA;;{0, 1}
7;6;Magazine;;{0}
7;8;rifleGen;out_triggerPressed;0
7;8;rifleGen;out_firingSelector;1
7;8;rifleGen;out_boltBack;0
7;8;rifleGen;out_magSeating;1
7;8;rifleGen;;{Messages Sent: 7, Sigma: 1, Test Phase: 1, Firing Mode: 1, Trigger Pressed: Yes, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
7;4;TA;;{inf, 0}
7;6;Magazine;out_bulletReady;1
7;6;Magazine;;{inf}
7;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
7;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
7;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
7;7;Bullet;out_isDud;0
7;7;Bullet;out_bulletReady;1
7;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
8;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
8;4;TA;;{0, 1}
8;6;Magazine;;{0}
8;8;rifleGen;out_triggerPressed;1
8;8;rifleGen;out_firingSelector;1
8;8;rifleGen;out_boltBack;0
8;8;rifleGen;out_magSeating;1
8;8;rifleGen;out_bulletLoaded;1
8;8;rifleGen;;{Messages Sent: 8, Sigma: 1, Test Phase: 1, Firing Mode: 1, Trigger Pressed: No, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
8;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
8;4;TA;out_releaseBolt;1
8;4;TA;;{inf, 0}
8;6;Magazine;out_bulletReady;1
8;6;Magazine;;{inf}
8;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
8;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
8;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
8;7;Bullet;out_isDud;0
8;7;Bullet;out_bulletReady;1
8;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
9;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
9;4;TA;;{0, 1}
9;6;Magazine;;{0}
9;8;rifleGen;out_triggerPressed;0
9;8;rifleGen;out_firingSelector;1
9;8;rifleGen;out_boltBack;0
9;8;rifleGen;out_magSeating;1
9;8;rifleGen;;{Messages Sent: 9, Sigma: 1, Test Phase: 1, Firing Mode: 1, Trigger Pressed: Yes, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
9;4;TA;;{inf, 0}
9;6;Magazine;out_bulletReady;1
9;6;Magazine;;{inf}
9;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 1}
9;2;Chbr;;{5} {dudBullet: 1, bulletIn: 1}
9;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
9;7;Bullet;out_isDud;1
9;7;Bullet;out_bulletReady;1
9;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 1}
10;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
10;4;TA;;{0, 1}
```

```
9;7;Bullet;out_bulletReady;1
9;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 1}
10;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
10;4;TA;;{0, 1}
10;6;Magazine;;{0}
10;8;rifleGen;out_triggerPressed;1
10;8;rifleGen;out_firingSelector;1
10;8;rifleGen;out_boltBack;0
10;8;rifleGen;out_magSeating;1
10;8;rifleGen;out_bulletLoaded;1
10;8;rifleGen;;{Messages Sent: 10, Sigma: 1, Test Phase: 1, Firing Mode: 2, Trigger Pressed: No, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
10;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
10;4;TA;out_releaseBolt;1
10;4;TA;;{inf, 0}
10;6;Magazine;out_bulletReady;1
10;6;Magazine;;{inf}
10;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
10;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
10;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
10;7;Bullet;out_isDud;0
10;7;Bullet;out_bulletReady;1
10;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
11;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
11;4;TA;;{0, 1}
11;6;Magazine;;{0}
11;8;rifleGen;out_triggerPressed;0
11;8;rifleGen;out_firingSelector;2
11;8;rifleGen;out_boltBack;0
11;8;rifleGen;out_magSeating;1
11;8;rifleGen;;{Messages Sent: 11, Sigma: 1, Test Phase: 1, Firing Mode: 2, Trigger Pressed: Yes, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10
11;4;TA;;{inf, 0}
11;6;Magazine;out_bulletReady;1
11;6;Magazine;;{inf}
11;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
11;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
11;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
11;7;Bullet;out_isDud;0
11;7;Bullet;out_bulletReady;1
11;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
12;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
12;4;TA;;{0, 1}
12;6;Magazine;;{0}
12;8;rifleGen;out_triggerPressed;1
12;8;rifleGen;out_firingSelector;2
12;8;rifleGen;out_boltBack;0
12;8;rifleGen;out_magSeating;1
12;8;rifleGen;out_bulletLoaded;1
12;8;rifleGen;;{Messages Sent: 12, Sigma: 1, Test Phase: 1, Firing Mode: 2, Trigger Pressed: No, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
12;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
12;4;TA;out_releaseBolt;1
12;4;TA;;{inf, 0}
12;6;Magazine;out_bulletReady;1
12;6;Magazine;;{inf}
12;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
12;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
12;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
12;4;TA;out_releaseBolt;1
12;4;TA;;{inf, 0}
12;6;Magazine;out_bulletReady;1
12;6;Magazine;;{inf}
12;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
12;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
12;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
12;7;Bullet;out_isDud;0
12;7;Bullet;out_bulletReady;1
12;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
13;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
13;4;TA;;{0, 1}
13;6;Magazine;;{0}
13;8;rifleGen;out_triggerPressed;0
13;8;rifleGen;out_firingSelector;2
13;8;rifleGen;out_boltBack;0
13;8;rifleGen;out_magSeating;1
13;8;rifleGen;;{Messages Sent: 13, Sigma: 1, Test Phase: 1, Firing Mode: 2, Trigger Pressed: Yes, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
13;4;TA;;{inf, 0}
13;6;Magazine;out_bulletReady;1
13;6;Magazine;;{inf}
13;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
13;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
13;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
13;7;Bullet;out_isDud;0
13;7;Bullet;out_bulletReady;1
13;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
14;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
14;4;TA;;{0, 1}
14;6;Magazine;;{0}
14;8;rifleGen;out_triggerPressed;1
14;8;rifleGen;out_firingSelector;2
14;8;rifleGen;out_boltBack;0
14;8;rifleGen;out_magSeating;1
14;8;rifleGen;out_bulletLoaded;1
14;8;rifleGen;;{Messages Sent: 14, Sigma: 1, Test Phase: 1, Firing Mode: 2, Trigger Pressed: No, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
14;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
14;4;TA;out_releaseBolt;1
14;4;TA;;{inf, 0}
14;6;Magazine;out_bulletReady;1
14;6;Magazine;;{inf}
14;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
14;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
14;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
14;7;Bullet;out_isDud;0
14;7;Bullet;out_bulletReady;1
14;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
15;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
15;4;TA;;{0, 1}
15;6;Magazine;;{0}
15;8;rifleGen;out_triggerPressed;0
15;8;rifleGen;out_firingSelector;2
15;8;rifleGen;out_boltBack;0
15;8;rifleGen;out_magSeating;1
15;8;rifleGen;;{Messages Sent: 15, Sigma: 1, Test Phase: 1, Firing Mode: 0, Trigger Pressed: Yes, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
```

```
15;8;rifleGen;out_magSeating;1
15;8;rifleGen;;{Messages Sent: 15, Sigma: 1, Test Phase: 1, Firing Mode: 0, Trigger Pressed: Yes, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
15;4;TA;;{inf, 0}
15;6;Magazine;out_bulletReady;1
15;6;Magazine;;{inf}
15;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
15;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
15;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
15;7;Bullet;out_isDud;0
15;7;Bullet;out_bulletReady;1
15;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
16;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
16;4;TA;;{0, 1}
16;6;Magazine;;{0}
16;8;rifleGen;out_triggerPressed;1
16;8;rifleGen;out_firingSelector;0
16;8;rifleGen;out_boltBack;0
16;8;rifleGen;out_magSeating;1
16;8;rifleGen;out_bulletLoaded;1
16;8;rifleGen;;{Messages Sent: 16, Sigma: 1, Test Phase: 1, Firing Mode: 0, Trigger Pressed: No, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
16;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
16;4;TA;out_releaseBolt;0
16;4;TA;;{inf, 0}
16;6;Magazine;out_bulletReady;1
16;6;Magazine;;{inf}
16;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
16;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
16;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
16;7;Bullet;out_isDud;0
16;7;Bullet;out_bulletReady;1
16;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
17;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
17;4;TA;;{0, 1}
17;6;Magazine;;{0}
17;8;rifleGen;out_triggerPressed;0
17;8;rifleGen;out_firingSelector;0
17;8;rifleGen;out_boltBack;0
17;8;rifleGen;out_magSeating;1
17;8;rifleGen;;{Messages Sent: 17, Sigma: 1, Test Phase: 1, Firing Mode: 0, Trigger Pressed: Yes, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
17;4;TA;;{inf, 0}
17;6;Magazine;out_bulletReady;1
17;6;Magazine;;{inf}
17;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
17;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
17;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
17;7;Bullet;out_isDud;0
17;7;Bullet;out_bulletReady;1
17;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
18;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
18;4;TA;;{0, 1}
18;6;Magazine;;{0}
18;8;rifleGen;out_triggerPressed;1
18;8;rifleGen;out_firingSelector;0
18;8;rifleGen;out_boltBack;0
18;8;rifleGen;out_magSeating;1
18;6;Magazine;;{0}
18;8;rifleGen;out_triggerPressed;1
18;8;rifleGen;out_firingSelector;0
18;8;rifleGen;out_boltBack;0
18;8;rifleGen;out_magSeating;1
18;8;rifleGen;out_bulletLoaded;1
18;8;rifleGen;;{Messages Sent: 18, Sigma: 1, Test Phase: 1, Firing Mode: 0, Trigger Pressed: No, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
18;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
18;4;TA;out_releaseBolt;0
18;4;TA;;{inf, 0}
18;6;Magazine;out_bulletReady;1
18;6;Magazine;;{inf}
18;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
18;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
18;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
18;7;Bullet;out_isDud;0
18;7;Bullet;out_bulletReady;1
18;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
19;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
19;4;TA;;{0, 1}
19;6;Magazine;;{0}
19;8;rifleGen;out_triggerPressed;0
19;8;rifleGen;out_firingSelector;0
19;8;rifleGen;out_boltBack;0
19;8;rifleGen;out_magSeating;1
19;8;rifleGen;;{Messages Sent: 19, Sigma: 1, Test Phase: 1, Firing Mode: 0, Trigger Pressed: Yes, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
19;4;TA;;{inf, 0}
19;6;Magazine;out_bulletReady;1
19;6;Magazine;;{inf}
19;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
19;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
19;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
19;7;Bullet;out_isDud;0
19;7;Bullet;out_bulletReady;1
19;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
20;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
20;4;TA;;{0, 1}
20;6;Magazine;;{0}
20;8;rifleGen;out_triggerPressed;1
20;8;rifleGen;out_firingSelector;0
20;8;rifleGen;out_boltBack;0
20;8;rifleGen;out_magSeating;1
20;8;rifleGen;out_bulletLoaded;1
20;8;rifleGen;;{Messages Sent: 20, Sigma: 1, Test Phase: 1, Firing Mode: 1, Trigger Pressed: No, Bolt Back: No, Mag Seated: Yes, Bullets Remaining: 10}
20;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
20;4;TA;out_releaseBolt;0
20;4;TA;;{inf, 0}
20;6;Magazine;out_bulletReady;1
20;6;Magazine;;{inf}
20;7;Bullet;;{0, State: ACTIVE, bulletRdy: 1, isDud: 0}
20;2;Chbr;;{5} {dudBullet: 0, bulletIn: 1}
20;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
20;7;Bullet;out_isDud;0
20;7;Bullet;out_bulletReady;1
20;7;Bullet;;{inf, State: PASSIVE, bulletRdy: 1, isDud: 0}
21;3;BA;;{inf, State: PASSIVE, boltFree: 0, readyBullet: 1, boltState: 0}
```



The output of the 3 scenarios is shown in the screenshots above. They Rifle model was tested using a generator to simulate the different scenarios. The logs confirm that the generator is successfully producing the specified scenarios (trigger toggles, magazine seating toggles, bolt toggles) and that the system's components respond as intended.

Initial Messages and State:

The system starts in Test Phase 1, with the firing mode initially set to 0 (safe) and the trigger not pressed.

Messages are generated at regular intervals, causing transitions between PASSIVE and ACTIVE states.

Scenario 1 Behavior:

Every internal transition, the trigger press status toggles (on/off).

Every 5 messages, the firing mode cycles (safe → single → auto → back to safe).

The log lines show where the firing mode changes and the trigger toggles accordingly.

Scenario 2 Behavior:

After enough messages are sent in Scenario 1, the generator transitions to Scenario 2.

Here, the magazine seating state toggles every 6 messages, and whenever the magazine is seated and bullets remain, the trigger is pressed (leading to a fire action).

The logs show when the magazine is seated or unseated, and how the bullet count decrements when a shot is fired.

Scenario 3 Behavior:

Finally, the generator switches to scenario 3, where the bolt position toggles each time.

If the bolt is back and there are bullets left, the trigger is pressed and a bullet is consumed. Otherwise, the trigger is not pressed.

The logs show these toggles and how the system transitions accordingly, with bullets decreasing when the bolt is back and the trigger is fired.

The simulation ends once the maximum number of messages is reached or no bullets remain.

Conclusion

This report outlined the implementation, testing, and evaluation of a discrete-event simulation model using the DEVS formalism. Through systematic testing of atomic and coupled models, the simulation successfully demonstrated correct state transitions, input processing, and output generation. Overall, the simulation provided a realistic representation of the rifle system using Cadmium DEVS modeling, highlighting the effectiveness of discrete-event simulations for analyzing complex cyber-physical systems