

```
*****
```

## Module

GenerateStationMessages.c

## Revision

1.0.1

## Description

This c file creates the messages which will be sent to the transmit service.

## Notes

## History

When	Who	What/Why
05/20/20	hbf	Wrote GenerateSonarMessage and GenerateConnMessage
05/19/20	dmc	Wrote GenerateTorpedoMessage

```
*****/
```

```
#include "DataParserService.h"
#include "ES_Configure.h" /* gets us event definitions */
#include "ES_Types.h" /* gets bool type for returns */
#include <stdio.h>
```

```
static int TurnsUntilDetonated = 2;
static struct GameStatus *GameStatusP;
static bool gameStatusPointerInitialized = false;
```

```
//Increment NumSpacesAway
static bool TorpedoEnRoute = false;
```

```
*****
```

## Function

GenerateTorpedoMessage

## Parameters

none

## Returns

uint32\_t message to send

## Description

This function generates a message for the Torpedo station

## Notes

## Author

D. Chan, 5/19/20

```
*****/
```

```
uint32_t GenerateTorpedoMessage(void){
    if(!gameStatusPointerInitialized) {
        GameStatusP = getPointerToGameStatus();
        gameStatusPointerInitialized = true;
    }
    uint32_t MessageToSend = ConvertToPacket(ENEMY, TORPEDO, NO_ACTION); //By default our message is no action
    //printf("No message: %d \r\n", MessageToSend);
    if (TurnsUntilDetonated == 1) TurnsUntilDetonated--;
    else if (TurnsUntilDetonated == 0)
    {
        MessageToSend = ConvertToPacket(ENEMY, TORPEDO, DETONATE);
        printf("DETONATED: %d \r\n", MessageToSend);
        TurnsUntilDetonated = 2;
    }
    else if (GameStatusP->WeLaunched == true) //If Fired = True, send fired, set fired false, set TurnsUntilDetonated
    {
        MessageToSend = ConvertToPacket(ENEMY, TORPEDO, LAUNCHED);
```

```

printf("LAUNCHED: %d \r\n", MessageToSend);
uint8_t X = GameStateP->OurPositionX;
uint8_t Y = GameStateP->OurPositionY;
uint8_t Dist = GameStateP->DistanceToDetonate;
if (GameStateP->OurHeading == UP)
{
    Y+=Dist;
}
else if (GameStateP->OurHeading == LEFT_UP)
{
    X-=Dist;
    Y+=Dist;
}
else if (GameStateP->OurHeading == RIGHT_UP)
{
    X+=Dist;
}
else if (GameStateP->OurHeading == LEFT_DOWN)
{
    X-=Dist;
}
else if (GameStateP->OurHeading == RIGHT_DOWN)
{
    X+=Dist;
    Y-=Dist;
}
else if (GameStateP->OurHeading == DOWN)
{
    Y-=Dist;
}
uint8_t packet = (Y<<4) + X;
SetStrikeLocation(packet);
printf("Set strike location to X: %d, Y: %d", X, Y);
GameStateP->WeLaunched = false;
TurnsUntilDetonated--;
}

return MessageToSend;
}

/*****
Function
    GenerateConnMessage

Parameters
    none

Returns
    uint32_t message to send

Description
    This function generates a message for the Conn station

Notes

Author
    H. Francis, 5/20/20
*****/
uint32_t GenerateConnMessage(void){
    if(!gameStatusPointerInitialized) {
        GameStateP = getPointerToGameState();
        gameStatusPointerInitialized = true;
    }
    uint32_t MessageToSend;
    if (GameStateP->NextConnMove == MOVE_FORWARD)
    {
        uint8_t H = GameStateP->OurHeading;
        if (H == UP)
        {

```

```

    GameStateP->OurPositionY++;
}
else if (H == LEFT_UP)
{
    GameStateP->OurPositionX--;
    GameStateP->OurPositionY++;
}
else if (H == RIGHT_UP)
{
    GameStateP->OurPositionX++;
}
else if (H == LEFT_DOWN)
{
    GameStateP->OurPositionX--;
}
else if (H == RIGHT_DOWN)
{
    GameStateP->OurPositionX++;
    GameStateP->OurPositionY--;
}
else if (H == DOWN)
{
    GameStateP->OurPositionY--;
}
MessageToSend = ConvertToPacket(OWN,CONN,POS_UPDATE);
} else if (GameStateP->NextConnMove == TURN_CCW || GameStateP->NextConnMove == TURN_CW) {
    bool ccw = true;
    if(GameStateP->NextConnMove == TURN_CW) ccw = false;

    uint8_t H = GameStateP->OurHeading;
    if (H == UP)
    {
        if(ccw)GameStateP->OurHeading = LEFT_UP;
        if(!ccw)GameStateP->OurHeading = RIGHT_UP;
    }
    else if (H == LEFT_UP)
    {
        if(ccw)GameStateP->OurHeading = LEFT_DOWN;
        if(!ccw)GameStateP->OurHeading = UP;
    }
    else if (H == RIGHT_UP)
    {
        if(ccw)GameStateP->OurHeading = UP;
        if(!ccw)GameStateP->OurHeading = RIGHT_DOWN;
    }
    else if (H == LEFT_DOWN)
    {
        if(ccw)GameStateP->OurHeading = DOWN;
        if(!ccw)GameStateP->OurHeading = LEFT_UP;
    }
    else if (H == RIGHT_DOWN)
    {
        if(ccw)GameStateP->OurHeading = RIGHT_UP;
        if(!ccw)GameStateP->OurHeading = DOWN;
    }
    else if (H == DOWN)
    {
        if(ccw)GameStateP->OurHeading = RIGHT_DOWN;
        if(!ccw)GameStateP->OurHeading = LEFT_DOWN;
    }
    MessageToSend = ConvertToPacket(OWN,CONN,POS_UPDATE);
} else {
    MessageToSend = ConvertToPacket(ENEMY,CONN,NO_ACTION);
}
return MessageToSend;
}

```

```

/*****

```

Function

GenerateSonarMessage

Parameters

none

Returns

uint32\_t message to send

Description

This function generates a message for the Sonar station

Notes

Author

H. Francis, 5/20/20

\*\*\*\*\*/

```
uint32_t GenerateSonarMessage(void){
    if(!gameStatusPointerInitialized) {
        GameStatusP = getPointerToGameStatus();
        gameStatusPointerInitialized = true;
    }
    uint32_t MessageToSend;
    if (GameStatusP->WeDirectPinged)
    {
        MessageToSend = ConvertToPacket(ENEMY,SONAR,PING_DIRECTED);
    } else if (GameStatusP->WeOmniPinged) {
        MessageToSend = ConvertToPacket(ENEMY,SONAR,PING_OMNI);
    } else {
        MessageToSend = ConvertToPacket(ENEMY,SONAR,NO_ACTION);
    }
    return MessageToSend;
}
```