# CS/EE 120B Custom Laboratory Project Report

# Space War


## Zerong Cai


## June 12th, 2024

**Introduction**

Space War is a shooting game with the theme being in space. When the game is being played, aliens will drop down from the screen and the player has to shoot at the aliens. Each kill will be recorded on the scoreboard on the screen, where each kill made by the player will be recorded as 1 point. To win the game, the player needs to achieve a total score of 20 points or kill 20 aliens, where the game can be restarted after that. Each alien also has different life points every time they come down, so the player has to shoot a different number of times at each alien in order to kill it and make it disappear. However, if an alien coming down the screen reaches the end of the screen, this indicates that the alien has escaped and was not shot, which results in a loss and the player's score will drop to 0. All important aspects of the game were implemented and all major bugs and errors that may limit usability were all fixed.

**Build-upons**

Passive Buzzer

My first build-upon is the passive buzzer. It was fully implemented where it was used to play four different songs for the game plus a sound effect for shooting.

LCD Graphics Display

My second build-upon is the LCD Graphics Display, where it is used to display the game. It was fully implemented where only characters related to the game were included on the display, such as sprites for the scoreboard, player, and the aliens.

IR Remote

My third build-upon is the IR remote, where it is also fully implemented. The remote is used for turning the game on and off. When the player presses the pause button, it turns the game on. When the player presses the power button, it turns the game off.

**User Guide**

The controls for the video game include the IR remote, joystick, and buttons. These will be used as inputs for the game. The IR remote is used to turn the game on and off. The joystick is used to control the spaceship, which represents the player for the game. It can only be moved left and right, so only the x-axis of the joystick is used. Finally, there are also three buttons. One button is the on/start button. When pressed the first time, it turns the graphics display by showing the game. When pressed a second time, it starts the game and initiates aliens to come down. The joystick button is also used to restart the game back to default whenever the player loses or wins the game. It can also be pressed during the game as well. The other button is used for shooting, where if pressed, the game will output a bullet from the spaceship.

The visual outputs of the game would be the spaceship at the bottom of the screen, green aliens with red eyes coming down from screen, bullets shooting out of the spaceship, and the score next to the characters “Score: ” at the top left corner of the display. The default settings of the game is when the spaceship is in the middle and the characters on the display, where aliens will fall down after pressing the button to start the game. The spaceship can be moved left or right using the joystick, where its bounds are the left and right side of the screen to prevent it from moving out of the screen. In addition to the graphics display, the 16x2 lcd display also outputs messages to the player as well. When the game is turned on using the IR remote, it will display “SPACE WAR” or the name of the game to the player. When a player loses, it will

display a “You Suck!!” message to the player. When a player wins, it will display a “Good Job!!” message to the player. When the game is turned off, the 16x2 lcd display and the graphics display along with the game will both turn blank, indicating that the game is off.

There are multiple sound mechanisms for the project as well. There are four songs played by the passive buzzer plus a shooting sound effect for the bullets. The first song is played when the player first presses the ON button to show the display of the game. This song will be played continuously until the player presses the button again to start the game. The second song is played when the player presses the same button to start the game. The third song is played when a player loses and the fourth is played when a player wins the game.The shooting sound effect is played whenever the player presses the shoot button.

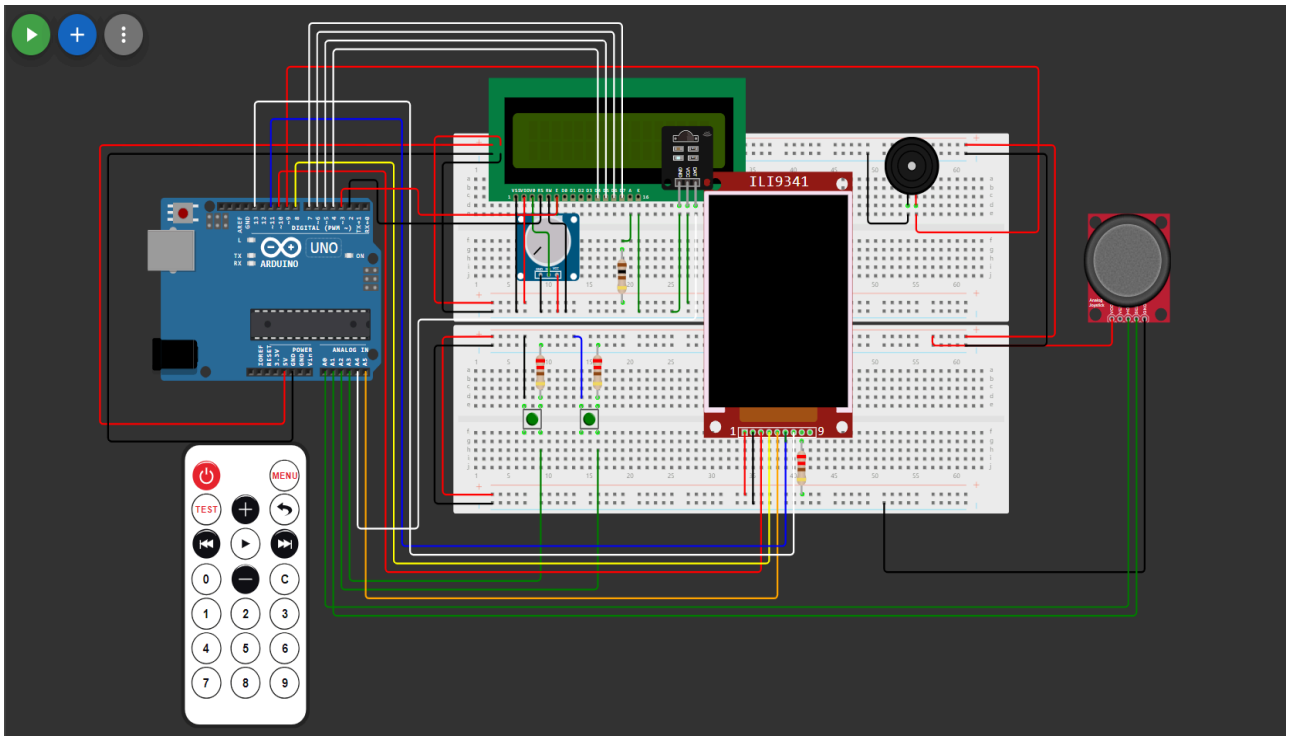
**Hardware Components Used**

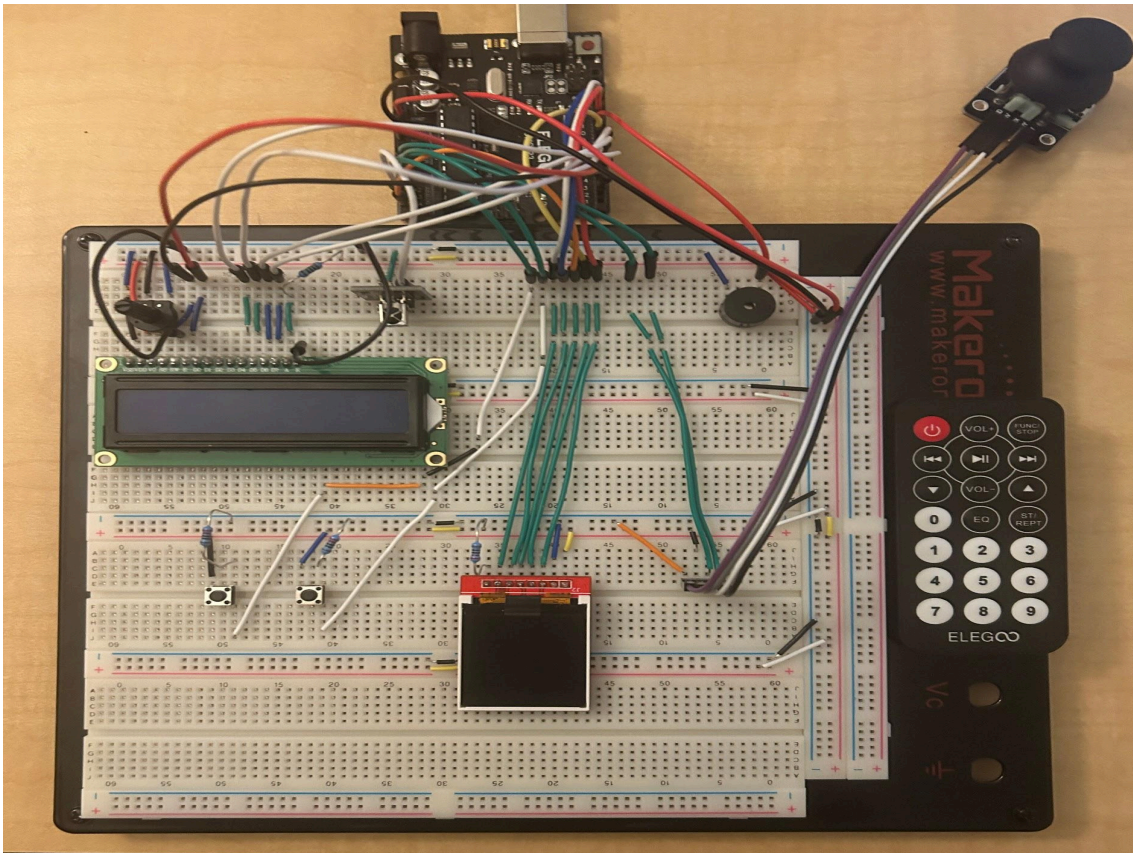
- 1 Graphics LCD display
- 1 IR Remote
- 1 IR Receiver Module
- 1 passive buzzer
- 2 buttons
- 1 Joystick
- 1 joystick button
- 1 16x2 LCD Display
- 1 potentiometer
- 3x220Ω resistors and 1x100Ω resistor

**Software Libraries Used**

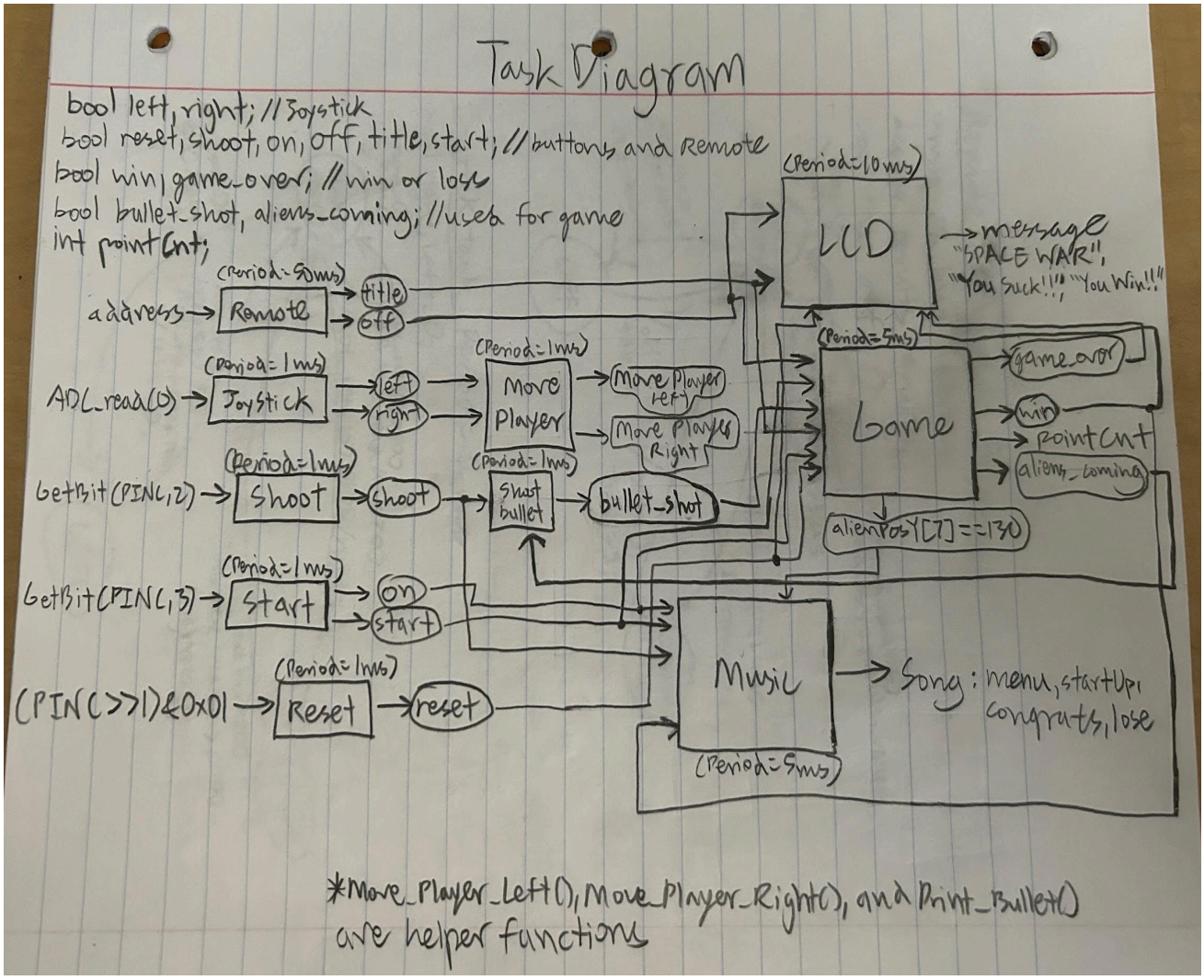
No external software libraries were used.

## Wiring Diagram

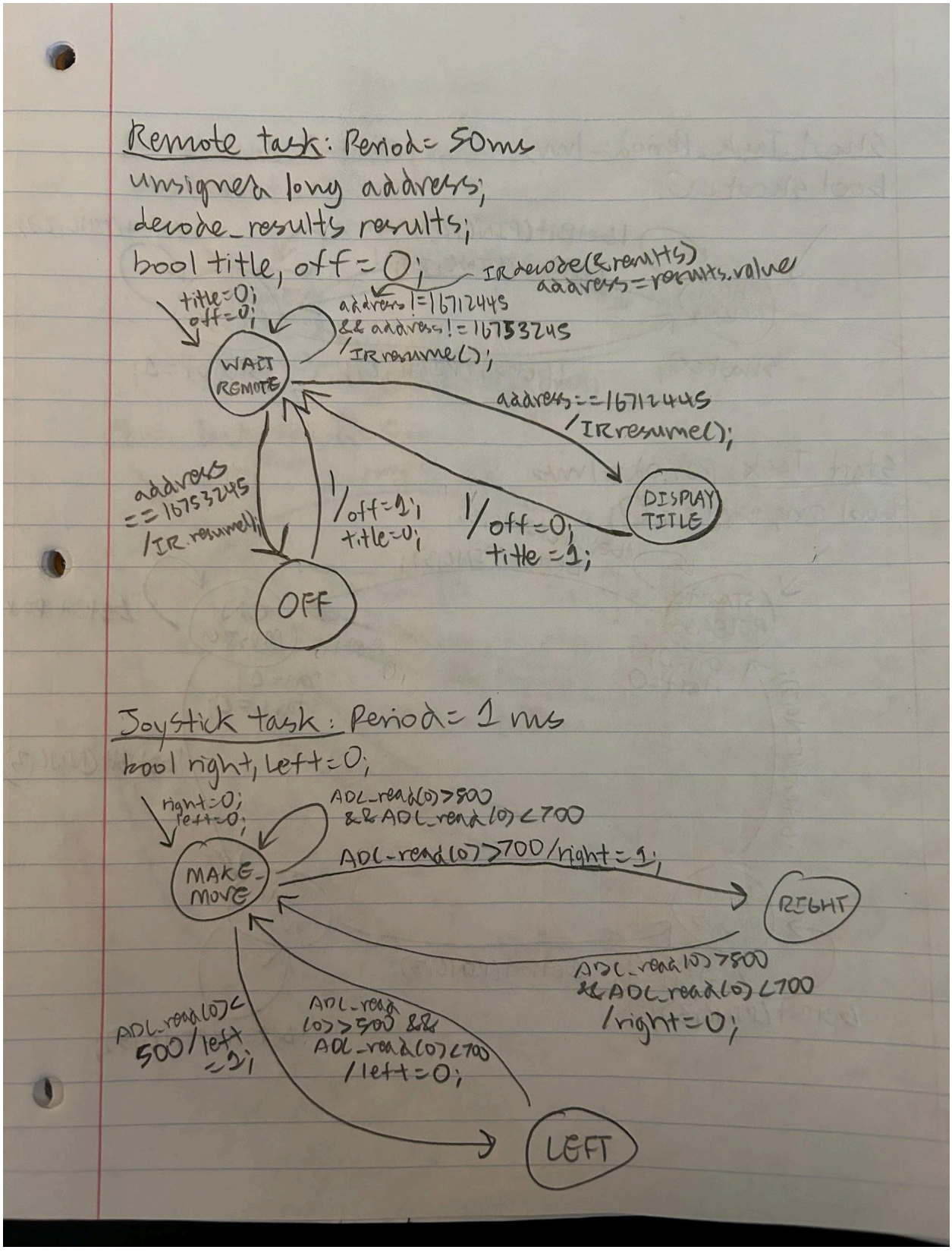


**Task Diagram**

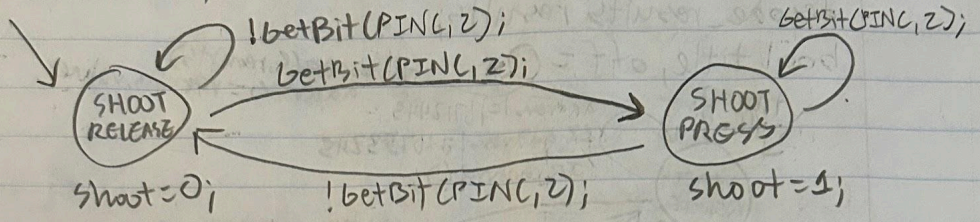


**SynchSM Diagrams**



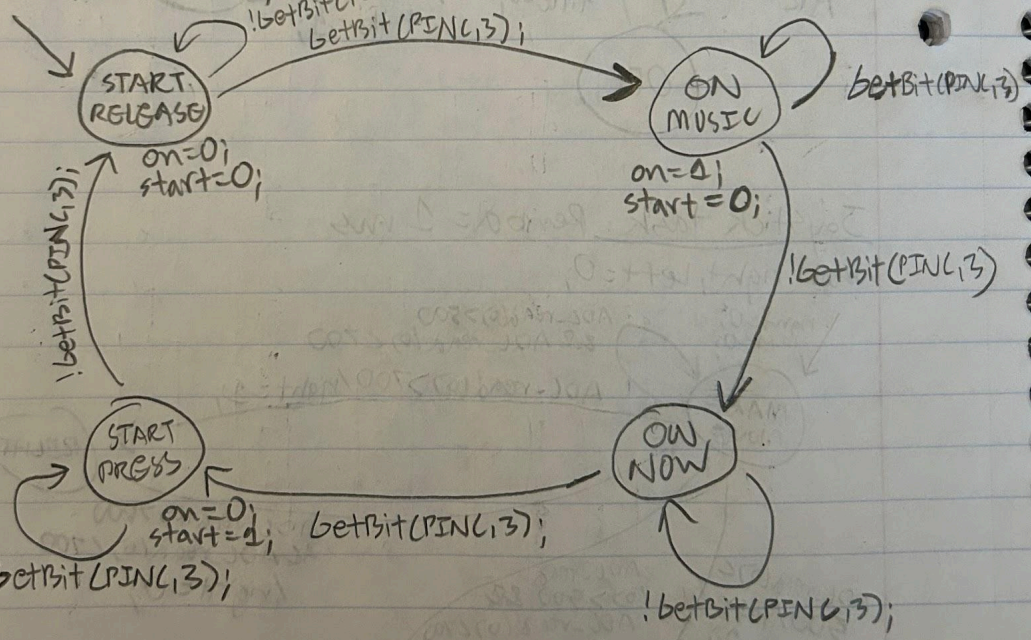
Shoot Task: Period = 1ms

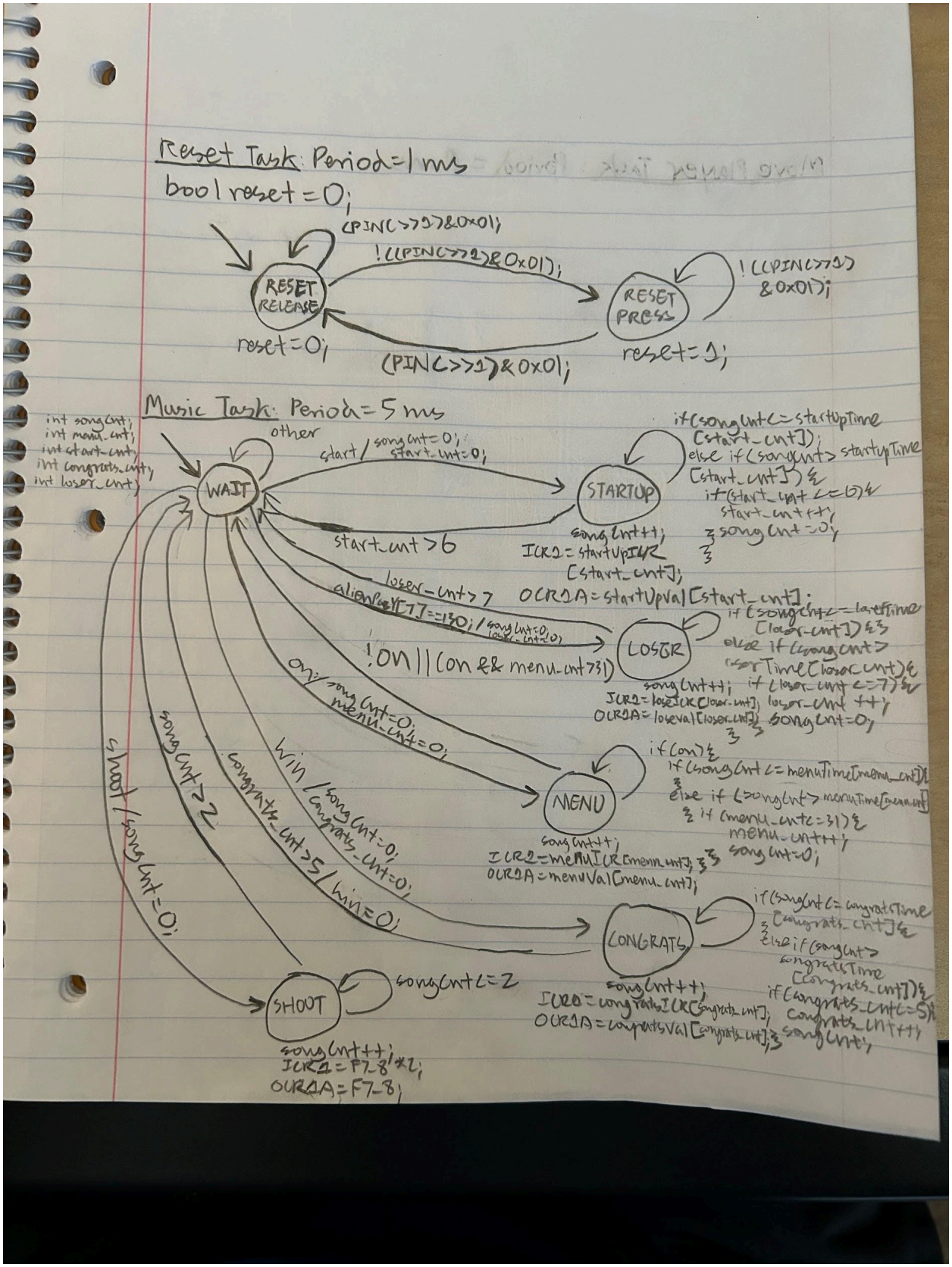
bool shoot = 0;



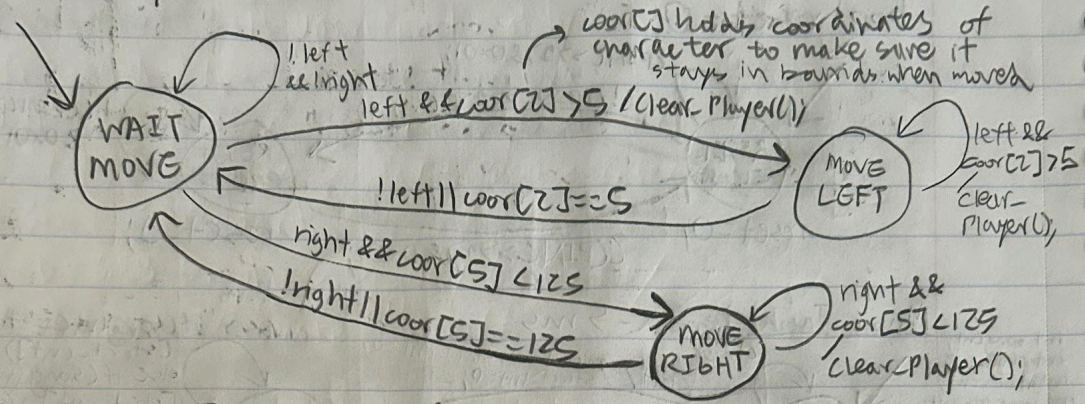

Start Task: Period = 1ms

bool on, start = 0;



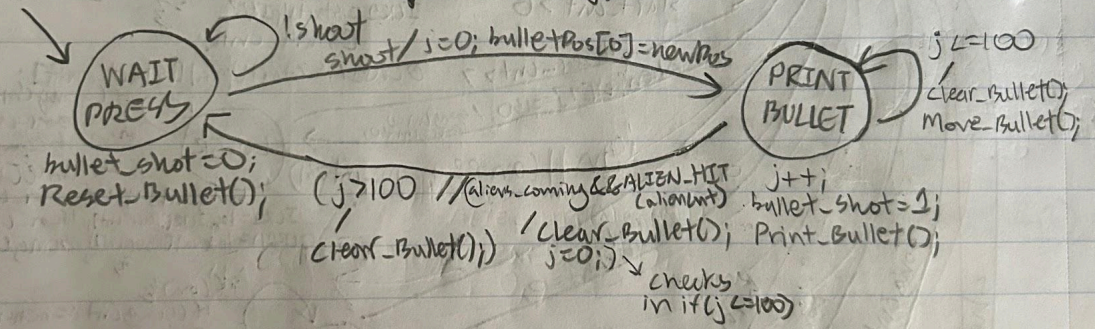

Reset Task: Period=1 ms
bool reset=0;
(PINC>>7)&0x01);
!((PINC>>7)&0x01);
RESET RELEASE
reset=0;
RESET PRESS
reset=1;
!((PINC>>7) &0x01);
(PINC>>1)&0x01);
Music Task: Period=5 ms
int songCnt;
int menu_cnt;
int start_cnt;
int congrats_cnt;
int loser_cnt;
other
WAIT
start / songCnt=0; start_cnt=0;
STARTUP
songCnt++;
start_cnt>6
losser_cnt>7
alienPos[7]==30; / songCnt=0; loser_cnt=0;
LOSER
!on || (on && menu_cnt>3)
on; / songCnt=0; menu_cnt=0;
MENU
songCnt>2
win / songCnt=0;
congrats_cnt=0;
congrats_cnt>5 / win=0;
CONGRATS
shoot / songCnt=0;
SHOOT
songCnt<2
songCnt++;
ICR1=F7_8/*2;
OCR1A=F7_8;
if(songCnt<=startupTime[start_cnt]);
else if(songCnt>startupTime[start_cnt]) {
if(start_cnt<6) start_cnt++;
songCnt=0;
}
ICR1=startupICR[start_cnt];
OCR1A=startupVal[start_cnt];
if(songCnt<=loserTime[loser_cnt]) {}
else if(songCnt>loserTime[loser_cnt]) {
if(loser_cnt<7) loser_cnt++;
songCnt=0;
}
ICR1=loseICR[loser_cnt];
OCR1A=loseVal[loser_cnt];
if(on) {
if(songCnt<=menuTime[menu_cnt]) {}
else if(songCnt>menuTime[menu_cnt]) {
if(menu_cnt<31) menu_cnt++;
songCnt=0;
}
ICR1=menuICR[menu_cnt];
OCR1A=menuVal[menu_cnt];
if(songCnt<=congratsTime[congrats_cnt]) {}
else if(songCnt>congratsTime[congrats_cnt]) {
if(congrats_cnt<5) congrats_cnt++;
songCnt=0;
}
ICR1=congratsICR[congrats_cnt];
OCR1A=congratsVal[congrats_cnt];

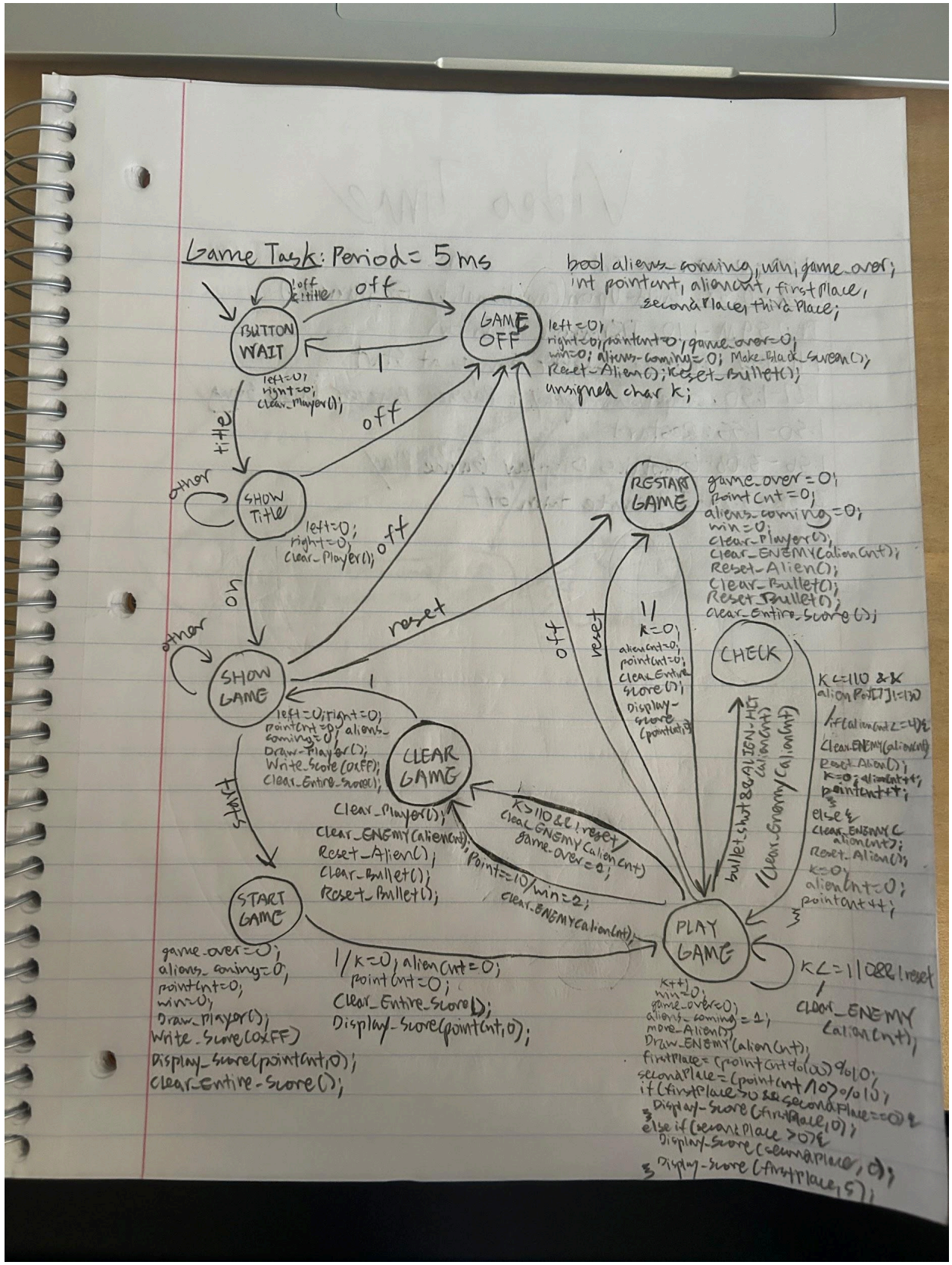
## Move Player Task: Period = 1ms

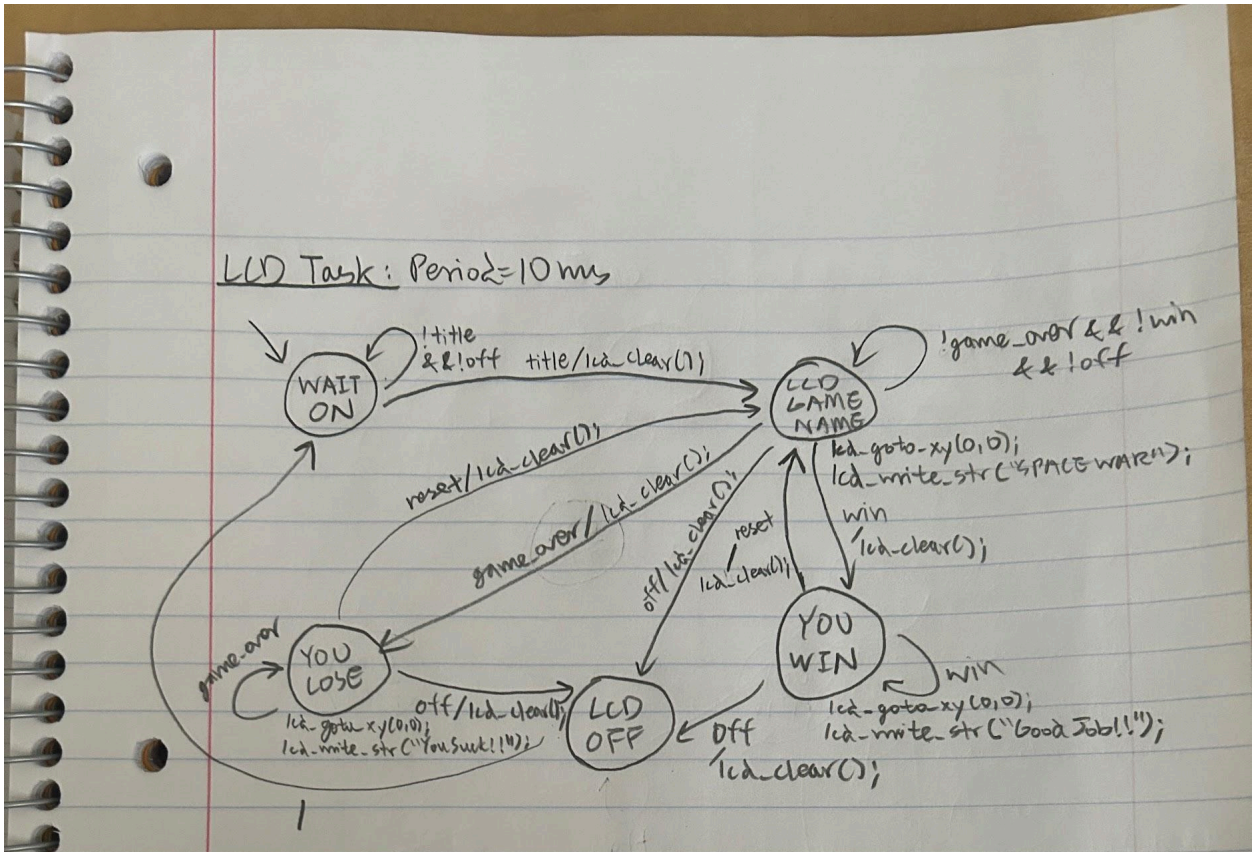



## Shoot Bullet Task: Period = 1ms

bool bullet_shot=0; unsigned char j;




Game Task: Period = 5 ms
bool aliens_coming, win, game_over;
int pointCnt, alienCnt, firstPlace, secondPlace, thirdPlace;
BUTTON WAIT
!off && title
off
GAME OFF
left=0; right=0; pointCnt=0; game_over=0; win=0; aliens_coming=0; Make_Black_Screen(); Reset_Alien(); Reset_Bullet(); unsigned char k;
1
left=0; right=0; Clear_Player();
title
off
other
SHOW Title
left=0; right=0; Clear_Player();
off
on
RESTART GAME
game_over=0; pointCnt=0; aliens_coming=0; win=0; Clear_Player(); Clear_ENEMY(alienCnt); Reset_Alien(); Clear_Bullet(); Reset_Bullet(); Clear_Entire_Score();
other
SHOW GAME
reset
off
reset
1
left=0; right=0; pointCnt=0; aliens_coming=0; Draw_Player(); Write_Score(0xFF); Clear_Entire_Score();
1
k=0; alienCnt=0; pointCnt=0; Clear_Entire_Score(); Display_Score(pointCnt);
CHECK
K<=110 && K alienPoint[]!=130
if(alienCnt<=4){ Clear_ENEMY(alienCnt); Reset_Alien(); k=0; alienCnt++; pointCnt++; } else { Clear_ENEMY(alienCnt); Reset_Alien(); k=0; alienCnt=0; pointCnt++; }
bullet shot && ALIEN_HIT (alienCnt)
!(Clear_Enemy(alienCnt))
start
CLEAR GAME
Clear_Player(); Clear_ENEMY(alienCnt); Reset_Alien(); Clear_Bullet(); Reset_Bullet();
K>110 && !reset
Clear_ENEMY(alienCnt); game_over=0;
Point==10 win=2; Clear_ENEMY(alienCnt);
START GAME
game_over=0; aliens_coming=0; pointCnt=0; win=0; Draw_Player(); Write_Score(0xFF); Display_Score(pointCnt,0); Clear_Entire_Score();
1
k=0; alienCnt=0; pointCnt=0; Clear_Entire_Score(); Display_Score(pointCnt,0);
PLAY GAME
K++; win=0; game_over=0; aliens_coming=1; moveAlien(); Draw_ENEMY(alienCnt); firstPlace=(pointCnt%10)%10; secondPlace=(pointCnt/10)%10; if(firstPlace>0 && secondPlace==0){ Display_Score(firstPlace,0); } else if(secondPlace>0){ Display_Score(secondPlace,0); Display_Score(firstPlace,5); }
K<=11 && !reset
Clear_ENEMY(alienCnt);


LCD Task: Period=10ms
WAIT ON
!title && !off
title/lcd_clear();
LCD GAME NAME
!game_over && !win && !off
lcd_goto_xy(0,0);
lcd_write_str("SPACE WARS");
win
/lcd_clear();
reset/lcd_clear();
game_over/lcd_clear();
off/lcd_clear();
reset
lcd_clear();
YOU WIN
win
lcd_goto_xy(0,0);
lcd_write_str("Good Job!!");
YOU LOSE
game_over
lcd_goto_xy(0,0);
lcd_write_str("You Suck!!");
off/lcd_clear();
LCD OFF
off
/lcd_clear();