File: C:\Users\TheAv\Desktop\Robot Code\3946X-2018-19\auton.c

```
//This file contains specific autonomous routiens

// Auton to run in square closest to flag
int autonTime=0;
void nearAutonFirstHalf(int side, bool flipCap) {
    startTask(clawTask);
    clawPID.target=clawClosePos;
    startTask(liftControl);
    liftPID.target=liftOutOfTheWayHeight;
    if(flipCap)liftPID.target-=100;
    startTask(rotatorTask);
    rotatorPID.target=rotatorLowPos;
    motor[intake]=127;
    if(flipCap) pDrive(-1150);//back up to get first ball
    else pDrive(-930);
    motor[intake]=0;
    if(flipCap) pDrive(1190);//drive back to starting square
    else if(side==BLUESIDE)pDrive(1050);
    else pDrive(920);
    SensorValue[gyro]=0;
    motor[slingshot]=127;
    pTurn(850*side);//turn to align with flag
    motor[slingshot] = 0;
    motor[intake]=127;
    wait1msec(400);
    motor[slingshot] = 127;
    pDrive(-650);//align with second flag
    wait1Msec(150);//shoot second fllag
    motor[slingshot]=0;
    if(side==BLUESIDE)pTurn(170*side);
    else pTurn(170*side);//align with bottom flag


}

void nearAutonPark(int side) {
    clearTimer(T1);
    nearAutonFirstHalf(side, false);
    pDrive(-550);//hit bottom flag
    if(side==BLUESIDE)pTurn(0)
        else pTurn(-150*side,false);

    if(side==BLUESIDE)pDrive(1800);//drive to align with park
    else pDrive(1850);
    pTurn(800*side);//turn to park

    drive(127);
    wait1Msec(1400);
    drive(0);

    //**/

    autonTime=time1[T1];

}

void nearAutonCap(int side) {
    clearTimer(T1);
    nearAutonFirstHalf(side, false);
```

```
    liftPID.target+=100;
    SensorValue[gyro]=0;
    pDrive(-550);//hit bottom flag

    pDrive(590);
    liftPID.target=500;
    motor[intake]=0;
    pTurn(900*side, false);//turn to face cap
    clawPID.target=clawOpenPos-200;
    drive(-127);
    wait1Msec(500);//back up on wall to align
    drive(0);
    SensorValue[gyro] = 0;


    pDrive(570);
    clawPID.target=clawClosePos;//grab cap
    wait1Msec(200);
    liftPID.target+=300;
    rotatorPID.target=rotatorHighPos;//flip cap
    wait1Msec(500);
    liftPID.target-=300;
    clawPID.target=clawOpenPos;
    pDrive(-500);//drive back
    autonTime = time1[t1];
}

void prog() {
    clearTimer(T1);
    nearAutonFirstHalf(REDSIDE, true);
    liftPID.target+=100;
    SensorValue[gyro]=0;
    pDrive(-600);//hit bottom flag
    pTurn(-200);

    pDrive(630);
    liftPID.target=500;
    motor[intake]=0;
    pTurn(900, false);//turn to align with wall
    drive(-127);
    wait1Msec(500);//back up to align on wall
    drive(0);
    SensorValue[gyro] = 0;

    clawPID.target=clawOpenPos-200;
    pDrive(570);
    clawPID.target=clawClosePos;//grab cap
    wait1Msec(200);
    liftPID.target+=300;
    pTurn(-900,false);//turn to get cap out of the way



    startTask(rotatorTask);
    rotatorPID.target=rotatorHighPos;//flip cap
    wait1Msec(500);
    liftPID.target-=300;
    clawPID.target=clawOpenPos;
    pDrive(-200);
```

```
    clawPID.target=clawClosePos;
    wait1Msec(250);
    pTurn(900,false);
    clawIdle = true;
    pDrive(775);
    pTurn(-900,false);
    liftPID.target+=100;//lift to avoid burning out intake
    motor[intake]=127;
    drive(-127);//back up to align on wall
    wait1Msec(1000);
    drive(0);
    SensorValue[gyro]=0;//line up on wall
    rotatorPID.target=rotatorLowPos;
    pDrive(700);
    liftPID.target=liftOutOfTheWayHeight;
    motor[intake]=127;
    motor[slingshot]=127;//shoot middle flag
    pTurn(-220);
    wait1Msec(500);
    motor[slingshot]=0;
    motor[intake]=0;
    pTurn(975);//line up with next cap
    clawIdle=false;
    liftPID.target=500;
    clawPID.target=clawOpenPos;
    pDrive(1000);
    clawPID.target=clawClosePos;//grab cap
    wait1Msec(500);
    liftPID.target+=400;
    pTurn(-900);
    rotatorPID.target=rotatorHighPos;//flip third cap
    wait1Msec(750);
    clawPID.target=clawOpenPos-200;
    liftPID.target-=400;
    pDrive(-200);
    pTurn(900);
    pDrive(230);
    pTurn(-900);
    drive(-127);//back up to hit third bottom flag
    wait1Msec(1000);
    drive(0);
    SensorValue[gyro]=0;

    //get to platform and park
    clawPID.target=clawClosePos;

    pDrive(500);
    pTurn(900);
    pDrive(300);
    pTurn(-900);
    pDrive(1600);
    liftPID.target=700;
    wait1Msec(200);
    pTurn(-900);//turn to align with park


    //get on lower platform
    drive(30);
```

```
    wait1Msec(600);
    drive(127);
    wait1Msec(1100);

    //get on higher platform
    drive(30);
    wait1Msec(800);
    drive(127);
    wait1Msec(1050);
    drive(-127);
    wait1Msec(100);
    drive(0);
    //pTurn(900);
    //pDrive(1000);

    autonTime=time1[T1];
}


void farAutonFirstHalf(int side) {
//red is 1, blue is -1
    clearTimer(T1);
    startTask(clawTask);
    clawPID.target=clawClosePos;
    startTask(liftControl);
    if(side==BLUESIDE)pTurn(610*side);
    else pTurn(570 * side);
    motor[slingshot] = 127;
    if(side==BLUESIDE)pDrive(110);
    else pDrive(120);
    wait1Msec(400);
    motor[slingshot] = 0;
    pDrive(-100);
    if(side==BLUESIDE)pTurn(-610*side);
    else pTurn(-580 * side);
    clawIdle = false;
    clawPID.target=clawOpenPos;
    startTask(rotatorTask);
    if(side==BLUESIDE) rotatorPID.target=rotatorHighPos;
    else rotatorPID.target=rotatorLowPos;
    motor[intake] = 127;
    if(side==BLUESIDE) pDrive(-900);
    else pDrive(-870);//back up to get first ball
    liftPID.target=500;//lower lift
    //pDrive(-270);
    if(side==BLUESIDE) pDrive(130);
    else pDrive(100);

    if(side==BLUESIDE) pTurn(1100 * side);
    else pTurn(1000*side);
    motor[intake]=0;
    liftPID.target=550;
    pDrive(380);
    clawPID.target=clawClosePos;//grab cap
    wait1Msec(400);
    liftPID.target+=300;
    if (side==BLUESIDE) rotatorPID.target=rotatorLowPos;
    else rotatorPID.target=rotatorHighPos;//flip cap
    wait1Msec(500);
```

```
    liftPID.target-=300;
    if(side==BLUESIDE)pTurn(-160*side,false,500);
    motor[intake] = 127;
    clawPID.target=clawOpenPos;
    getLiftOutOfTheWay();
    liftPID.target+=100;
}

// Auton to run in sqaure farthest to flag
void farAuton(int side) {
//red is 1, blue is -1
    farAutonFirstHalf(side);
    drive(-127);//back up to align on bar
    wait1Msec(800);
    motor[intake]=0;
    drive(-50);
    wait1Msec(300);
    SensorValue[gyro]=0;
    pDrive(150);
    motor[intake]=127;
    if(side==BLUESIDE)pTurn(-150*side,false,250);
    else pTurn(-200*side,false,250);
    motor[slingshot]=127;
    if(side==BLUESIDE) pDrive(80);
    else pDrive(0);
    wait1Msec(350);
    motor[slingshot]=0;
    drive(-127);
    if(side==BLUESIDE) wait1Msec(1410);
    else wait1Msec(1420);
    drive(127);
    wait1Msec(100);
    drive(0)

    autonTime=time1[T1];
}
```