```
#include "main.h"

void liftMonitorTask(void *parameter){
    while(true){
        mutexTake(runLiftMutex, 100);
        bool run = runLift;
        mutexGive(runLiftMutex);

        mutexTake(liftTicksMutex, 100);
        int target = liftTargetTicks;
        mutexGive(liftTicksMutex);

        while(run){
            bool needsToLower = (abs(encoderGet(liftQuad)) > target) ? true : false;
            switch(needsToLower){
                case true:
                    while(abs(encoderGet(liftQuad)) > target){
                        dLift(needsToLower);
                        delay(20);
                    }
                    stopLift();
                    run = false;
                    break;
                case false:
                    while(abs(encoderGet(liftQuad)) < target){
                        dLift(needsToLower);
                        delay(20);
                    }
                    stopLift();
                    run = false;
                    break;
            }

            mutexTake(runLiftMutex, 100);
            runLift = false;
            mutexGive(runLiftMutex);

        }

        delay(20);
    }
}

void setSyncLift(int targetTicks){
    mutexTake(liftTicksMutex, 100);
    liftTargetTicks = targetTicks;
    mutexGive(liftTicksMutex);

    mutexTake(runLiftMutex, 100);
    runLift = true;
    mutexGive(runLiftMutex);
}

void dLift(bool down){
    mutexTake(motorReqMutex, 100);
    motorReq[upperLift - 1] = down ? LIFT_POWER : -LIFT_POWER;
    motorReq[lowerRightLift - 1] = down ? LIFT_POWER : -LIFT_POWER;
    motorReq[lowerLeftLift - 1] = down ? -LIFT_POWER : LIFT_POWER;
    mutexGive(motorReqMutex);
}

void stopLift(){
    mutexTake(motorReqMutex, 100);
    motorReq[upperLift - 1] = 0;
    motorReq[lowerRightLift - 1] = 0;
    motorReq[lowerLeftLift - 1] = 0;
```

```
    mutexGive(motorReqMutex);

    mutexTake(motorMutexes[upperLift - 1], 100);
    motorStop(upperLift);
    mutexGive(motorMutexes[upperLift - 1]);

    mutexTake(motorMutexes[lowerRightLift - 1], 100);
    motorStop(lowerRightLift);
    mutexGive(motorMutexes[lowerRightLift - 1]);

    mutexTake(motorMutexes[lowerLeftLift - 1], 100);
    motorStop(lowerLeftLift);
    mutexGive(motorMutexes[lowerLeftLift - 1]);
}
```