```
#include "main.h"

void clawMonitorTask(void *parameter){
    while(true){
        mutexTake(downPressureMutex, -1);
        bool down = downPressure;
        mutexGive(downPressureMutex);

        mutexTake(runFingerMutex, -1);
        bool run = runFinger;
        mutexGive(runFingerMutex);

        mutexTake(clawClosingMutex, -1);
        bool closing = clawClosing;
        mutexGive(clawClosingMutex);

        bool open = (digitalRead(leftFingerSwitchPort) == 1 || digitalRead(rightFingerSwitchPort) == 1);

        if(closing == false){ //If in Auto and claw is not closing
            if(run == false){
                if(down == true){
                    mutexTake(motorMutexes[fingerY - 1], -1);
                    motorSet(fingerY, 15);
                    mutexGive(motorMutexes[fingerY - 1]);
                }else if(down == false){
                    mutexTake(motorMutexes[fingerY - 1], -1);
                    motorSet(fingerY, -5);
                    mutexGive(motorMutexes[fingerY - 1]);
                }else{
                    open = (digitalRead(leftFingerSwitchPort) == 1 || digitalRead(rightFingerSwitchPort) == 1);
                }
            }
        }

        while(run && open){
            mutexTake(motorMutexes[fingerY - 1], -1);
            motorSet(fingerY, -127);
            mutexGive(motorMutexes[fingerY - 1]);
            open = (digitalRead(leftFingerSwitchPort) == 1 || digitalRead(rightFingerSwitchPort) == 1);
            if(open == false || (joystickGetDigital(1, 5, JOY_DOWN) && !isAutonomous())){
                run = false;

                mutexTake(runFingerMutex, -1);
                runFinger = false;
                mutexGive(runFingerMutex);
            }
            delay(20);
        }
        delay(20);
    }
}

void closeClaw(int millis){
    mutexTake(downPressureMutex, -1);
    downPressure = true;
    mutexGive(downPressureMutex);

    mutexTake(runFingerMutex, -1);
    runFinger = false;
    mutexGive(runFingerMutex);

    mutexTake(clawClosingMutex, -1);
    clawClosing = true;
    mutexGive(clawClosingMutex);

    if(millis != 0){
        mutexTake(motorMutexes[fingerY - 1], -1);
        motorSet(fingerY, CLAW_POWER);
        mutexGive(motorMutexes[fingerY - 1]);

        mutexTake(motorReqMutex, -1);
        motorReq[fingerY - 1] = CLAW_POWER;
        mutexGive(motorReqMutex);

        delay(millis);

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

        mutexTake(clawClosingMutex, -1);
        clawClosing = false;
        mutexGive(clawClosingMutex);
    }else{
        mutexTake(motorMutexes[fingerY - 1], -1);
        motorSet(fingerY, CLAW_POWER);
        mutexGive(motorMutexes[fingerY - 1]);
    }

}

void openClaw(){

    mutexTake(clawClosingMutex, -1);
    clawClosing = false;
    mutexGive(clawClosingMutex);

    mutexTake(motorMutexes[fingerY - 1], -1);
    motorSet(fingerY, -CLAW_POWER);
    mutexGive(motorMutexes[fingerY - 1]);

    mutexTake(downPressureMutex, -1);
    downPressure = false;
    mutexGive(downPressureMutex);

    mutexTake(runFingerMutex, -1);
    runFinger = true;
    mutexGive(runFingerMutex);
}
```