

Anleitung zum Gebrauch von MultiSonArms 23-Feb-22.doc

17. Dez. 2020: Erstversion
10. Feb. 2021: Revision – Gebrauch von MultiSonArms aus dem GUI heraus siehe [2]
09. Apr. 2021: Erweiterung: Misslungenen Subprozesse reparieren mit [MultiSonArmsR](#).
13. Apr. 2021: Bessere Anleitung zum reparieren von misslungenen Subprozessen auf Seite 4.
23. Feb. 2022: [MultiSonArms.exe](#) und [MultiSonArmsR.exe](#) mit neuer Versionsnummer V1.4.0.
Keine Funktionsänderungen, basiert auf SonX 6.4.0.

Den Überblick, wozu MultiSonArms dient, ist gegeben in der Gebrauchsanleitung [2].

In dieser Anleitung hier wird beschrieben, wie MultiSonArms nicht nur aus dem GUI heraus, sondern auch von einem Batch-Programm aus aufgerufen werden kann, was jedoch nur in speziellen Fällen sinnvoll ist. Wenn Sie das nicht interessiert, genügt es, [2] zu lesen. Es ist einfacher bzw. benutzerfreundlicher, MultiSonArms aus dem SonArms GUI heraus zu nutzen, was in [2] erklärt wird. Da gibt es weitere Hinweise, die hier nicht nochmals aufgeführt sind.

Das kann [MultiSonArms.exe](#)

Im Unterschied zu [SonArms_Kernel](#) hat [MultiSonArms](#) folgende Besonderheiten:

- [MultiSonArms](#) kann seit dem 10. Feb. 2021 auch aus dem GUI heraus aufgerufen werden. Dies ist die empfohlene Betriebsart. Dafür brauchen Sie diese Anleitung hier nicht zu lesen.
- Es wird nur der Rastermodus (Programmparameter r für Lärmkarten) unterstützt. Denn es sind keine Projekte im Punktmodus bekannt, die besonders lange laufen würden.
- Der Punktmodus (Programmparameter p für Empfangspunkte) wird im Prinzip nicht unterstützt. Und doch, hier wird ein Projekt genauso in Subprojekte aufgeteilt, wie beim Rastermodus und sie werden auch als Subprozesse parallel ausgeführt. Es werden aber keine Subprojekt-Resultate zusammen gesetzt. Das könnte bei Bedarf manuell gemacht werden. Vorgesehen war, den Punktmodus gar nicht zu unterstützen. [MultiSonArms](#) meldet einen Fehler, auch wenn die Subprozesse erfolgreich ausgeführt worden sind. Ob letzteres der Fall ist, lässt sich am Ende leicht nachprüfen, indem die Logdateien und Statusdateien der Subprozesse überprüft werden.
- Mehr zu allfälligen Resultatunterschieden steht in [2].
- Ab 09. Apr. 2021 steht zusätzlich das Programm [MultiSonArmsR](#) zur Verfügung. Das GUI selber nutzt dieses Programm nicht. Es dient der Wiederherstellung von misslungenen Subprozessen und spart deswegen Zeit bei besonders lang laufenden Berechnungen. [MultiSonArmsR](#) fügt genau wie [MultiSonArms](#) Subprozess-Resultate zusammen, lässt aber alle Berechnungen und Subprozesse aus. Beide Programme werden völlig identisch aufgerufen. (Das [R](#) kommt von der READONLY oder RECOVER Vorstellung, wobei zwar auch hier Resultate komplettiert, also geschrieben werden.)

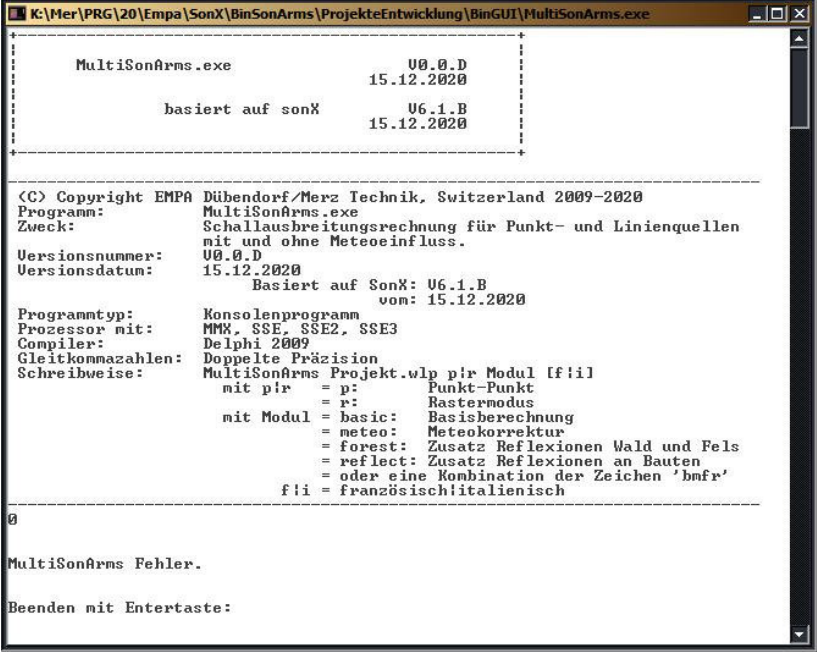
Die Installation von [MultiSonArms.exe](#)

[MultiSonArms](#) muss ins GUI Verzeichnis (üblicherweise [BinGUI](#)) kopiert werden. Sämtliche Begleitinformationen, die zum GUI und zum Rechenkern gegeben sind (und die von den Benutzenden im GUI festgelegt worden sind), stehen dadurch auch [MultiSonArms](#) zur Verfügung. [MultiSonArms](#) läuft zwar auch ganz ohne GUI, aber dann müssen die GUI Dateien [Version.txt](#), [Paths.txt](#), [PathsSonArms.txt](#) und alle in diesen Dateien notierten Dateien zur Verfügung stehen. Und sowieso auch der [SonArms_Kernel](#) mit allen Settings Dateien.

Ab 09. Apr. 2021 soll auch das Programm [MultiSonArmsR](#) ins GUI Verzeichnis kopiert werden.

Der Aufruf von **MultiSonArms.exe**

MultiSonArms wird auf die exakt gleiche Art aufgerufen wie der **SonArms_Kernel**. Ein Start des Programmes aus dem Explorer heraus (ohne Programmparameter) zeigt die Schreibweise, wie es richtig aufgerufen werden kann:



```
K:\Mer\PRG\20\Empa\SonX\BinSonArms\ProjekteEntwicklung\BinGUI\MultiSonArms.exe

MultiSonArms.exe          U0.0.D
                          15.12.2020

      basiert auf sonX      U6.1.B
                          15.12.2020

-----
(C) Copyright EMPA Dübendorf/Merz Technik, Switzerland 2009-2020
Programm:          MultiSonArms.exe
Zweck:             Schallausbreitungsrechnung für Punkt- und Linienquellen
                   mit und ohne Meteeinfluss.
Versionsnummer:    U0.0.D
Versionsdatum:     15.12.2020
                   Basiert auf SonX: U6.1.B
                   vom: 15.12.2020

Programmtyp:       Konsolenprogramm
Prozessor mit:     MMX, SSE, SSE2, SSE3
Compiler:          Delphi 2009
Gleitkommazahlen: Doppelte Präzision
Schreibweise:      MultiSonArms Projekt.wlp pir Modul [fli]
                   mit pir = p: Punkt-Punkt
                           = r: Rastermodus
                   mit Modul = basic: Basisberechnung
                           = meteo: Meteo Korrektur
                           = forest: Zusatz Reflexionen Wald und Fels
                           = reflect: Zusatz Reflexionen an Bauten
                           = oder eine Kombination der Zeichen 'bmfr'
                           fli = französisch/italienisch

0

MultiSonArms Fehler.

Beenden mit Entertaste:
```

Ein Aufruf kann z.B. wie folgt aussehen:

Programmpfad	Projektpfad	Modus
"..\BinGUI\MultiSonArms.exe"	"..\Projekte\P213 Versuche\SimplonKlein.wlp"	r basic

(Beide Pfade hier sind Relativpfade. MultiSonArms verträgt ab V1.0.2 aber auch absolute Pfade.)

Relative Pfade setzen voraus:

- Programmpfad:** Ausgehend vom gerade aktuellen Verzeichnis geht es erst 2 Verzeichnisse zurück. Dann geht es wieder vorwärts nach **BinGUI**, wo **MultiSonArms** platziert ist.
- Projektpfad:** Ausgehend vom Rechenkernverzeichnis (typisch **Bin**) geht es erst 2 Verzeichnisse zurück. Dann geht es vorwärts nach **Projekte** und weiter vorwärts nach **P213 Versuche**, wo dann die Projektdatei **SimplonKlein.wlp** platziert ist.
- Modus:** Der Arbeitsmodus hier ist der Rastermodus (Lärmkarte) mit dem basic Modul.

Die Anführungszeichen sind nötig wegen allfälligen Leerzeichen in den Pfaden. Wäre im obigen Beispiel der Projektpfad ohne Anführungszeichen, wäre wegen dem Leerzeichen die Anzahl Programmparameter um 1 erhöht, was völlig falsche, d.h. unverständliche Fehler-Botschaften provozieren würde. Aufpassen muss man auch mit Umlauten in den Pfaden oder Dateinamen. **MultiSonArms** unterstützt sie zwar, aber in **.bat** Dateien gilt ein anderer Zeichensatz.

Die Ausführungsschritte von **MultiSonArms.exe**

Nacheinander führt **MultiSonArms** die folgenden Schritte aus:

Beginn: **MultiSonArms** wird wie oben angegeben mit dem Projektpfad (**.wlp**) als Programmparameter gestartet. Aus dem GUI Verzeichnis holt sich **MultiSonArms** die Information (aus **Version.txt**), welche Version des Rechenkerns zu benutzen ist und aus **Paths.txt**, in welchem Verzeichnis der Rechenkern sich befindet. Im Rechenkernverzeichnis liest das Programm zudem weitere Voreinstellungen aus der Datei **PathsSonArms.txt** aus. So hat es Zugriff auf alle vom Projekt genutzten Dateien.

- Löschen:** Im Projektverzeichnis befindet sich eventuell ein Verzeichnis **TempTasks**.
Das Verzeichnis TempTasks wird ohne Warnung komplett gelöscht!
In **TempTasks** enthalten sind alle Logdateien und Resultate eines früheren Laufs. Mit drin sind Kopien sämtlicher Settings und Inputs des aktuellen Projektes. *Wem z.B. die Logdateien später wichtig sind, der sollte diese Dateien manuell retten, bevor er MultiSonArms erneut anstösst.*
- Erstellen:** Im Projektverzeichnis wird das Verzeichnis **TempTasks** erstellt.
- Aufteilen:** **MultiSonArms** teilt das **.wlp** Originalprojekt in **n** Projekte/Prozesse auf und kopiert alles Nötige in die **n** Unterverzeichnisse (mit unserem Beispiel):
TempTasks\0001\SimplonKlein0001.wlp
TempTasks\0002\SimplonKlein0002.wlp
TempTasks\0003\SimplonKlein0003.wlp
usw.
- Das Originalprojekt (hier **SimplonKlein.wlp**) wird im wesentlichen kopiert, einzig die Quellen werden auf die Prozesse verteilt und Pfade im Projekt angepasst. Die Zahl der **n** Prozesse ist limitiert durch den **SonArms** Parameter **MaxThreads** oder durch die Zahl der tatsächlich vorhandenen CPUs. Die kleinere der beiden Zahlen bestimmt die tatsächliche Zahl der Prozesse. Ist die Zahl der Quellen im Projekt noch kleiner als das, bestimmt sie die tatsächliche Zahl der Prozesse. Mit **MaxThreads** kann also die Benutzerin festlegen, dass nicht alle CPUs genutzt werden sollen. Die Standardvorgabe des GUI ist gegenwärtig **MaxThreads** = 16, was zu wenig ist, wenn noch mehr CPUs genutzt werden sollen. Dieser Vorgabewert wird gelegentlich erhöht werden.
- .bat** **MultiSonArms** generiert die Dateien **MultiRun.bat** und **MultiKill.bat** und schreibt sie ins Verzeichnis **TempTasks**. Die Bedeutung dieser .bat Dateien ist:
MultiRun.bat Startet alle Subprozesse. **MultiSonArms** führt diese Datei nachher aus.
Oder die Benutzerin kann das bei Bedarf starten.
MultiKill.bat Stoppt alle Prozesse sofort. Das kann nötig werden bei einem Abbruch, wo die Prozesse fröhlich weiter laufen können, was lästig sein kann.
Der Benutzer kann das bei Bedarf auslösen.
- Start:** **MultiSonArms** startet alle Subprozesse durch Aufruf von **MultiRun.bat**.
Eine Statusdatei wird geschrieben.
- Warten:** **MultiSonArms** wartet auf das Ende aller Subprozesse und zeigt deren Ausführungsphase und Ausführungszustand an (Aktualisierung im Intervall von 10 Sekunden).
- Vereinigen:** **MultiSonArms** fügt die **.wlm** Resultate zusammen und speichert sie im ordentlichen Original-Output-Verzeichnis. Dieses Resultat sieht so aus, wie wenn das Originalprojekt alleine mit **SonArms_Kernel** gelaufen wäre und es steht gerade auch am richtigen Ort. Es steht dadurch für die Weiterverarbeitung durch **SonArmsLr** zur Verfügung (Berechnung der Beurteilungspegel).
- Ende ok:** Liegt kein Fehler vor, stoppt **MultiSonArms** seine Ausführung ohne Meldung. Wer den Moment verpasst hat und gerne sicher wäre was passiert ist, kann die Statusdatei lesen. Wer noch mehr wissen will, kann die einzelnen Status- und Logdateien der Subprozesse untersuchen.
- Ende nok:** Liegt ein Fehler vor, bleibt das **MultiSonArms** Fenster erhalten, wo die Botschaft steht, was falsch gelaufen ist. Wer noch mehr wissen will, kann die einzelnen Status- und Fehlerdateien der Subprozesse untersuchen.
- Beachte:** Fehler- Status- und Logdateien der Subprozesse unter **TempTasks** allenfalls retten vor der nächsten Ausführung von **MultiSonArms**, insbesondere bei Diagnosebedarf!

Die Nutzung von **MultiSonArmsR.exe** zusammen mit **MultiSonArms.exe**

Das Programm **MultiSonArmsR.exe** dient der Wiederherstellung von misslungenen Subprozessen, damit am Ende alle Resultate doch noch erfolgreich zu einem Endresultat zusammengesetzt werden können. Eine solche Wiederherstellung erfolgt nicht aus dem GUI heraus. Für die Instandstellung von Resultaten benutzen Sie die beiden folgenden **.bat** Dateien. Diese sind ans Projekt angepasst und werden von **MultiSonArms** (ab V1.2.0) bei jedem Aufruf immer wieder neu erzeugt:

- MultiRunSub.bat:** Teilt ein Subprojekt weiter auf und führt es parallelisiert aus. Typischerweise lassen Sie damit einen zuvor misslungenen Subprozess erneut ausführen.
- MultiAssemble.bat:** Fügt die Resultate eines Subprozesses zusammen und setzt das Resultat zu den Resultaten des übergeordneten Prozesses. Auf diese Weise ergänzen Sie (Teil-)Resultate um fehlende Resultatdateien.

Diese beiden **.bat** Dateien gehören ausschliesslich zum Kontext des aktuellen **.wlp** Projektes. In den **.bat** Programmen kommt der Projektname, aktuelle Verzeichnisse und der Arbeitsmodus des Rechenkerns vor. **Beim Neustart eines parallelisierten Laufs werden sie gelöscht.** Sie werden von den Benutzenden der Situation entsprechend eingesetzt und sollen mit Bedacht verwendet werden.

Nebeneffekt: Durch manuell-rekursiven Aufruf lässt sich so ein Projekt beliebig fein aufteilen in Sub-Sub-Sub-Sub-....Prozesse, was bei der Fehlersuche hilfreich sein kann.

Arbeitsschritte für den Fall eines abgebrochenen Subprozesses, den Sie reparieren möchten

- 1) **Startbedingung.** Sie haben Ihr Projekt im *Lärmkartenmodus parallelisiert* gestartet. Das erfolgt meistens aus dem GUI heraus.
- 2) **Abbruch.** Nachdem alle Subprozesse fertig sind, meldet **MultiSonArms**, dass eine Resultatdatei nicht kopiert werden kann. Das bedeutet, dass mindestens ein Subprozess abgebrochen wurde. Dem Dateipfad der Meldung entnehmen Sie die Nummer des abverheiten Subprozesses, z.B. 0003. Merken Sie sich diese Nummer.
- 3) **Sicherungskopie.** Wenn immer **MultiSonArms** abbricht, erstellen Sie sofort (insbesondere bei lang laufenden Projekten) eine Sicherungskopie des Projektes, falls Sie das gerne untersucht haben möchten. Denn das **TempTask** Verzeichnis wird bei einem erneuten Versuch gelöscht und es steht für Analysen oder Reparaturen nicht mehr zur Verfügung!
- 4) **Wechsle das Verzeichnis zum Subprozess unter TempTasks.** Schicken Sie die Maschine (z.B. im Explorer) in das Subprozess-Verzeichnis des abverheiten Subprozesses, das Sie sich bei 2) gemerkt haben. z.B. 0003. Wenn Sie die Nummer nicht mehr wissen, schauen Sie in **Out**, wo Resultate fehlen oder durch **.dat** Dateien auffallen. Die Nummer steht im Dateinamen. Im Subprozess-Verzeichnis (z.B. 0003) finden Sie das Programm **MultiRunSub.bat**.
- 5) **Start des Subprozesses.** Der Aufruf von **MultiRunSub.bat** startet eine parallelisierte Ausführung des Rechenkerns mit dem Subprojekt, hier 0003.
- 6) **Warten auf Erfolg.** Warten Sie auf die erfolgreiche Ausführung und Zusammensetzung der Sub-Sub-Prozesse zum Sub-Prozessresultat. Bei erneutem Nichterfolg liegt wahrscheinlich ein reproduzierbarer Fehler vor. Bitte senden Sie in dem Fall eine Kopie des gesamten Projektes inklusive **TempTasks** Ordner den Entwicklern.
- 7) **Wechsle das Verzeichnis zu TempTasks.** Schicken Sie die Maschine eine Stufe zurück. Dort befindet sich das Programm **MultiAssemble.bat**.
- 8) **Subprozessresultate zusammenführen.** Der Aufruf von **MultiAssemble.bat** startet die Zusammenführung der Subresultate zum Hauptresultat. Dieses steht ab jetzt für weitere Schritte mit dem GUI zur Verfügung, z.B. für die Berechnung des Beurteilungspegels.
- 9) **Misserfolg?** Bei Misserfolg kann auch noch ein anderer Subprozess misslungen gewesen sein. In dem Fall verfahren Sie mit letzterem gleich wie oben beschrieben. Ausser es wären zu viele Subprozesse ohne Erfolg verlaufen, dann lassen Sie besser das ganze Projekt richtig laufen.

Links:

- [1] SonArms Dokumentation im SonArms.zip Paket à Google mit „sonArms“ oder direkt:
<https://www.bafu.admin.ch/bafu/de/home/themen/laerm/fachinformationen/laermmittlung-und-beurteilung/ermittlung-und-beurteilung-von-schiesslaerm.html>
- [2] Gebrauchsanleitung des SonArms GUI mit MultiSonArms.doc