



































## 8 Appendix A: Script Example

**Listing 1.1.** Example user script for running automated capture and saving traces to MATLAB file

```
1 lst = [  
2 # Clock Routing Setup  
3   ['CW Extra', 'CW Extra Settings', 'Clock Source',  
4     'Target IO-IN'],  
5   ['OpenADC', 'Clock Setup', 'ADC Clock', 'Source',  
6     'EXTCLK x4 via DCM'],  
7 # Sample Length/Offset Setup  
8   ['OpenADC', 'Trigger Setup', 'Total Samples', 3000],  
9   ['OpenADC', 'Trigger Setup', 'Offset', 1500],  
10 # Low Noise Amplifier Gain Setting  
11  ['OpenADC', 'Gain Setting', 'Setting', 45],  
12 # Rising Edge Trigger  
13  ['OpenADC', 'Trigger Setup', 'Mode', 'rising edge'],  
14 # Final step: make DCMs relock in case they lost sync  
15  ['OpenADC', 'Clock Setup', 'Relock DCMs', None], ]  
16  
17 # cap variable contains instance of ChipWhispererCapture()  
18  
19 # Download all hardware setup parameters  
20 for cmd in lstexample: cap.setParameter(cmd)  
21  
22 # Set number of traces  
23 cap.setParameter(['Generic Settings', 'Acquisition Settings',  
24   'Number of Traces', 75])  
25  
26 # Capture a few traces initially (not saved)  
27 cap.capture1()  
28 # pe() is a macro which processes any queued events it must  
29 # be called when interacting with the low-level API directly.  
30 pe()  
31 cap.capture1()  
32 pe()  
33  
34 # Start capture process of 75 traces, save to memory  
35 writer = cap.captureM()  
36  
37 # Save files to MATLAB workspace file instead of native format  
38 sio.savemat('sca_data.mat', {'powertrace': writer.traces,  
39   'textin' : writer.textins,  
40   'textout' : writer.textouts,  
41   'knownkey' : writer.knownkey})
```