**CW305 Whitepaper:**

<https://www.newae.com/post/whitepaper-cw305>

**CW305 Target:**

<https://rtfm.newae.com/Targets/CW305%20Artix%20FPGA/>

**How to build a CW305 Project:**

<http://wiki.newae.com/Tutorial_CW305-1_Building_a_Project#Capture_Setup>

**CW305-ADC Lock Fail:**

1. Check if the power source is capable of driving both the capture board and CW305.

Reference: <https://forum.newae.com/t/cw-lite-and-cw305-adc-lock-fail/318>

**An example of CW305 Implementation:**

<https://github.com/markmatthewanderson/CESEL-CW305-test>

**Another Example, CW305-Arm DesignStart:**

<https://github.com/newaetech/CW305-Arm-DesignStart>

**How to adapt CW305:**

Start by making a copy of the AES example project. Remove the AES files and keep cw305\_top.v, cw305\_reg\_aes.v, and cw305\_usb\_reg\_fe.v.

Take cw305\_reg\_aes.v as a example and modify it for your own needs.

Then, see how CW305.py reads/writes registers in your FPGA design, via the fpga\_write() and fpga\_read() methods.

Make sure you understand how, in our reference design, the register interface is used to provide the “go” signal to the target, which in turn sets the IO4 line high which signals to the capture board to start capturing power samples.

Reference: <https://forum.newae.com/t/how-to-adapt-cw305-for-non-crypto-applications/2498/3>

**Related Files:**

<https://github.com/newaetech/chipwhisperer/blob/develop/software/chipwhisperer/capture/targets/CW305.py>

<https://github.com/newaetech/chipwhisperer/blob/49260f902e2018c931d3ad08f15c8b42092649fc/hardware/victims/cw305_artixtarget/fpga/common/cw305_reg_aes.v>