D-TACQ BOLO8-BLF Quickstart Guide

Sean Alsop
Contact: sean.alsop@d-tacq.co.uk

September 18, 2019

Contents

1 Introduction and pre-requisites **1**
1.1 Recommended reading 2

2 Hardware setup and confguration **2**

3 Using CS Studio to stream live data **2**

4 Calibrating the system **3**
4.1 Installing the acq400_hapi package 5
4.1.1 Hello world 5
4.2 Choosing channels to calibrate 6
4.3 Running a calibration 6

5 Performing the stream test after a calibration **6**
5.1 Taking a transient capture after a calibration 6

6 Conclusion **7**

7 Appendix: Embedded web page diagnostics **8**

1 Introduction and pre-requisites

This document serves as a quickstart guide to the BOLO8-BLF module. It will walk the user through the initial setup, configuration, and testing to make sure that the module is in working order. This document relies on the user already having Control System Studio (CSS) installed on their host computer. Installation instructions are provided on the D-TACQ ACQ400CSS GitHub repo here:
https://github.com/D-TACQ/ACQ400CSS
To control the unit there is a python script included in the acq400_hapi GitHub repository. The repo (and installation instructions) can be found here:
https://github.com/D-TACQ/acq400_hapi.
The script can be found in user_apps/special/ and is named “bolo8_cal_cap_loop.py”.

This script is not needed for immediate use, but is needed for system calibration and shot control later in the guide. The python calibration script (bolo8_cal_cap_loop.py) is also available on the UUT if the HAPI package is included. This is covered in section 4.

1.1 Recommended reading

1. The D-TACQ 4G User Guide
2. The BOLO8-BLF User Guide
3. The BOLO8-BLF Software Guide

2 Hardware setup and confguration

The D-TACQ BOLO8-BLF test panel uses a ZIF (zero insertion force) socket to hold the bolometer and allows the user to measure various test points on the bolometer when in use. This is the panel D-TACQ recommend using initially. Once the bolometer has been fitted to the test panel the user can connect the panel to the front of the BOLO8-BLF module. Figure 1 shows a bolometer sitting in the test panel ZIF socket. Figure 2 shows the same test board connected to the BOLO8-BLF mezzanine in the acq2106 carrier.

3 Using CS Studio to stream live data

CS Studio can be used to stream lower rate data from the acq2106. This is extremely useful for initial validation and testing as it allows the user to verify that the system is working very quickly. Once CS Studio has been installed according to the instructions contained in the GitHub repo the BOLO8 Launcher can be opened from the project window. There are several channels on the BOLO8-BLF module, but the voltage magnitude (MAG) works well for this test. Click on the MAG button in the "Live Plot" container, and also open the Capture OPI. To see the shark-fin shown in figure 4 there was a bicycle light placed on top of the bolometer. The shark-fin shape is the data acquired as the foils heat up and cool down as the continuously flashing bicycle lamp turns on and off. Note that the channels have not yet been calibrated and so the offsets have not been accounted for yet. This means that some channels likely will be visually offset from one another. In figure 4 this can be observed on the first three channels. Figure 5 shows the same data with channels masked down to CH01, allowing the display to autoscale on CH01.

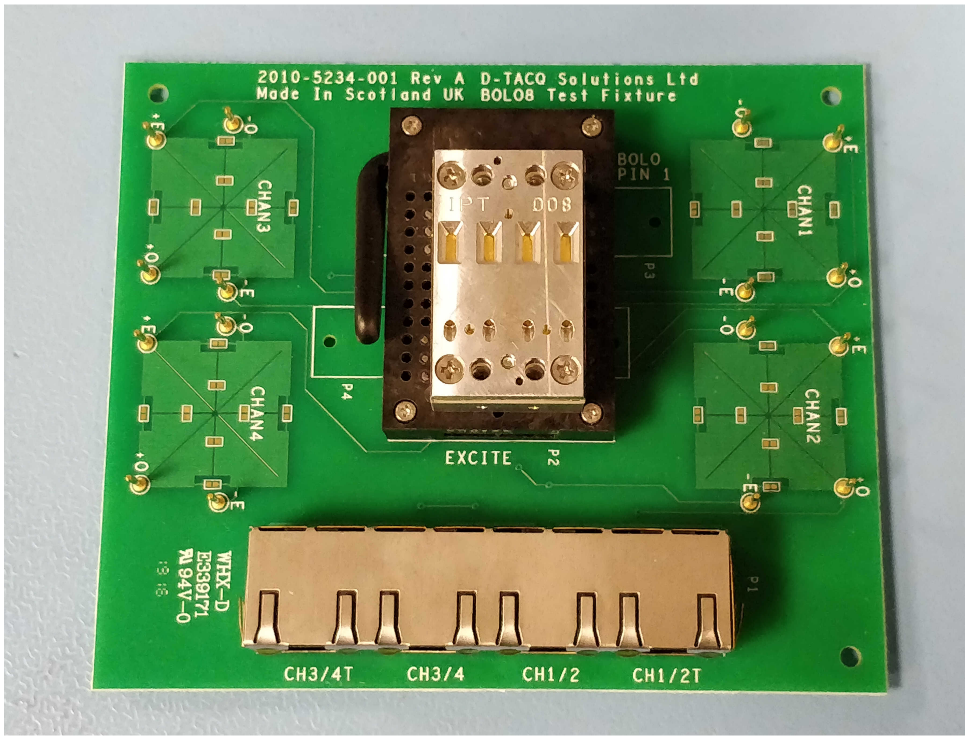

2010-5234-001 Rev A D-TACQ Solutions Ltd
Made In Scotland UK BOLO8 Test Fixture
BOLO PIN 1
CHAN1
CHAN2
CHAN3
CHAN4
EXCITE
CH3/4T
CH3/4
CH1/2
CH1/2T

Figure 1: A bolometer sitting in the test board ZIF socket

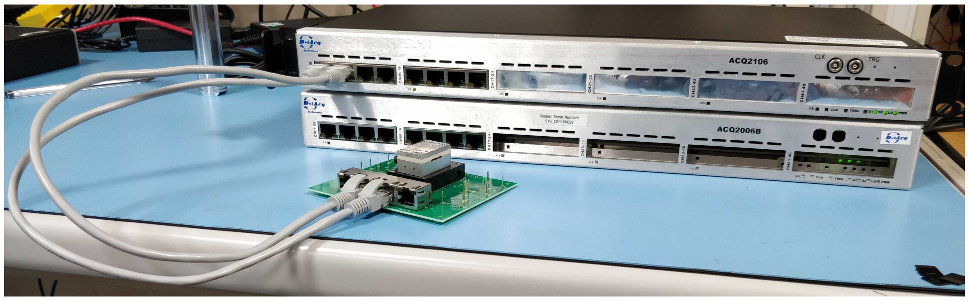


Figure 2: A bolometer sitting in the test board ZIF socket connected to an acq2106 system.

4 Calibrating the system

To calibrate the system it is recommended that the bolo8_cal_cap_loop.py python script is used.
This can be found in the HAPI python library provided by D-TACQ mentioned previously in this document.

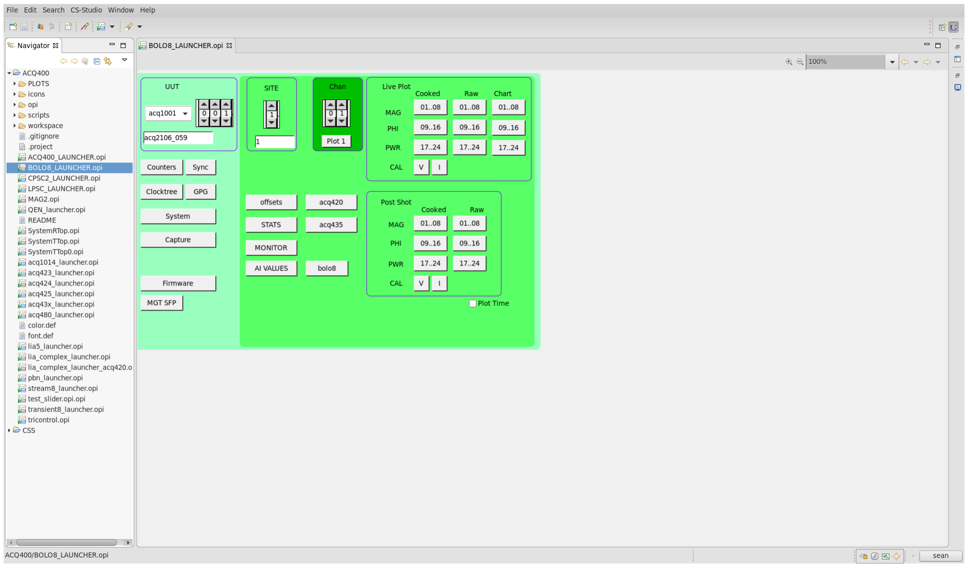


Figure 3: A CS Studio screen showing where to access the correct BOLO8 launcher.

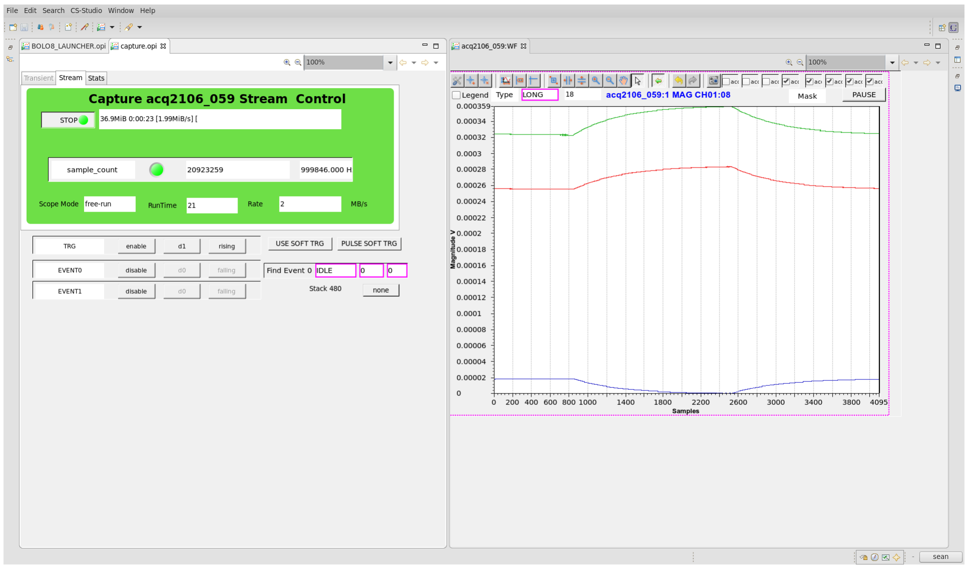
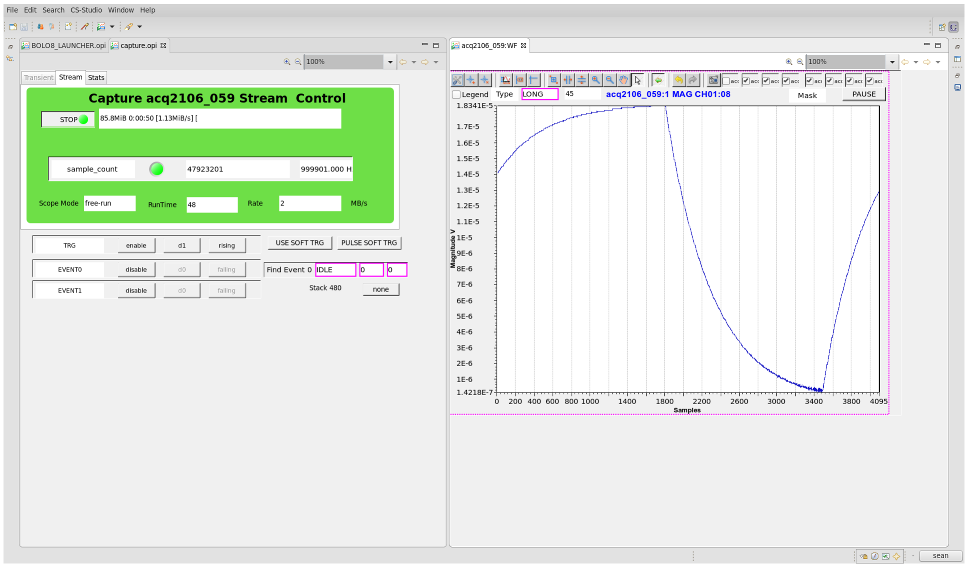


Figure 4: A CS Studio screen showing 3 channels without calibration. Note the offsets between channels.



Capture acq2106_059 Stream Control

85.8MiB 0:00:50 [1.13MiB/s] [

sample_count 47923201 999901.000 H

Scope Mode free-run RunTime 48 Rate 2 MB/s

TRG enable d1 rising

EVENT0 disable d0 falling

EVENT1 disable d0 falling

Find Event 0 IDLE 0 0

Stack 480 none

Type LONG 45 acq2106_059:1 MAG CH01:08

Figure 5: The capture and MAG stream OPIs open with a sharkfin on the MAG OPI.

4.1 Installing the acq400_hapi package

The acq400_hapi package can also be found on the acq2106 system provided the acq400_hapi package has been included. The user can include the acq400_hapi package (if it is not already installed) by doing the following from an ssh session on the acq2106:

```
cp /mnt/packages.opt/*hapi* /mnt/packages/
```

The system will need to be rebooted to make sure these changes are included. The following command should be used to reboot:

```
sync;sync;reboot
```

4.1.1 Hello world

The HAPI package installs HAPI to the following location on the acq2106:

```
/usr/local/HAPI/
```

With bolo8_cal_cap_loop.py being located here:

```
/usr/local/HAPI/user_apps/special/bolo8_cal_cap_loop.py
```

4.2 Choosing channels to calibrate

To calibrate first connect to the acq2106 using ssh and edit the following file:

```
/mnt/local/sysconfig/bolo.sh
```

Select which channels you wish to calibrate by changing the following line:

```
BOLO_ACTIVE_CHAN="1 2 3 4"
```

4.3 Running a calibration

Remember to remove the flashing bicycle lamp from the vicinity of the bolometer when calibrating the BOLO8-BLF module. Once the desired calibration channels have been chosen the python script can than be run from the host computer or from the acq2106 like so:

```
./bolo8_cal_cap_loop.py --cal=1 --cap=0 --shots=1 acq2106_123
```

5 Performing the stream test after a calibration

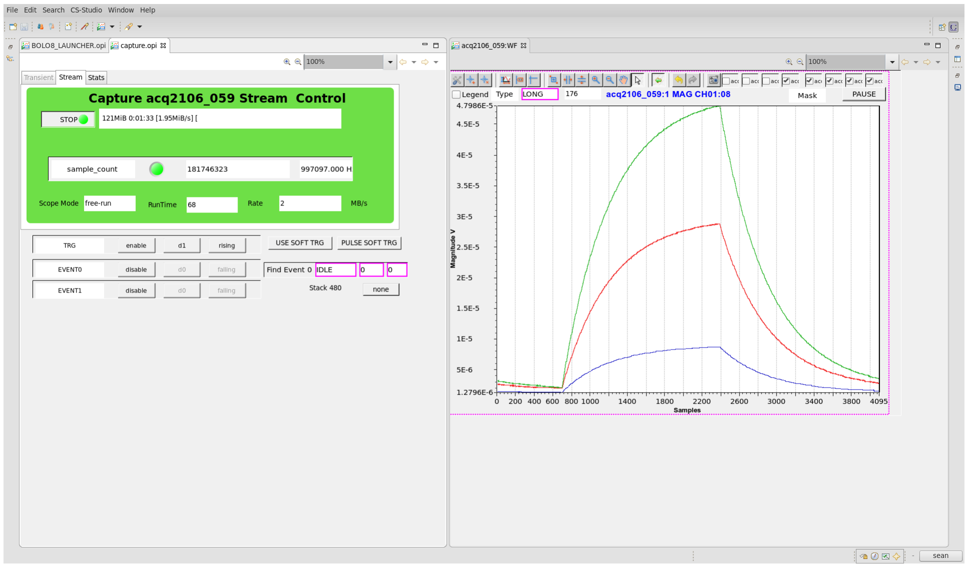
The same CS Studio test can be performed after having calibrated the system. Figure 6 only shows 3 fins because the bolometer at use for testing at D-TACQ only has 3 working foils. Note that all of the channels are now offset from the same value (very close to 0 volts). The fins having varying heights can be attributed to the torch being positioned at an angle above the bolometer and the light from it being distributed unevenly. Figure 7 shows the offset current and voltages for the calibration run. This allows the user to verify that nothing has gone wrong with the calibration. Channel 4 on figure 7 is an example of no offset current applied.

5.1 Taking a transient capture after a calibration

After taking a calibration a transient capture can also be taken to capture the flashing light. This can be done using the bolo8_cal_cap_loop.py script like so:

```
./bolo8_cal_cap_loop.py --cal=0 --cap=1 --shots=1 acq2106_123
```

This will show several bike lamp flashes over 100k samples. At a default sample rate of 10kHz the capture will take 10 seconds. Open the CS Studio BOLO8 post shot MAG plot from the BOLO8 launcher shown in figure 3. The data should look like figure 8



Capture acq2106_059 Stream Control

121MiB 0:01:33 [1.95MiB/s] [

sample_count 181746323 997097.000 H

Scope Mode free-run RunTime 68 Rate 2 MB/s

TRG enable d1 rising

EVENT0 disable d0 falling

EVENT1 disable d0 falling

Find Event 0 IDLE 0 0

Stack 480 none

Figure 6: A CS Studio screen showing 3 channels after a calibration has been performed. Note the offsets have disappeared.

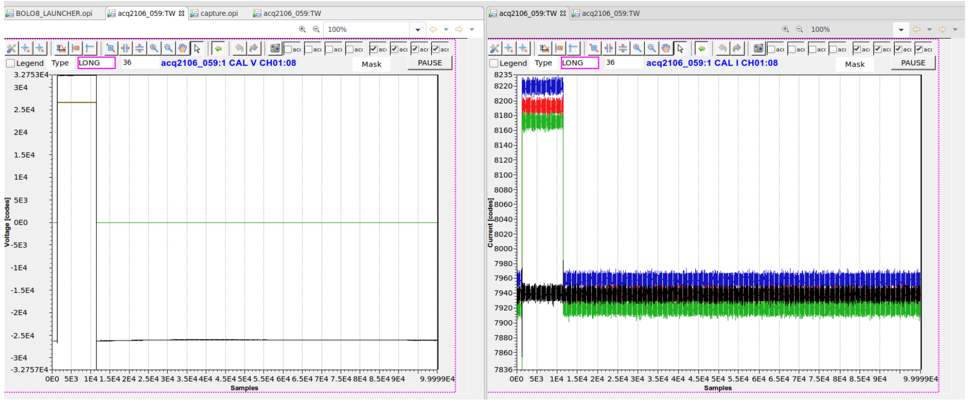


Figure 7: A CS Studio screen showing 3 channels of offset voltage and offset current after a calibration has been performed. Note the broken bolometer channel.

6 Conclusion

This document has illustrated the recommended method for testing a D-TACQ acq2106 fitted with the BOLO8-BLF mezzanine. The recommended steps are to insert the bolometer in the D-TACQ bolomter test panel ZIF socket and connect the panel to the acq2106. Once CS Studio is installed the user can

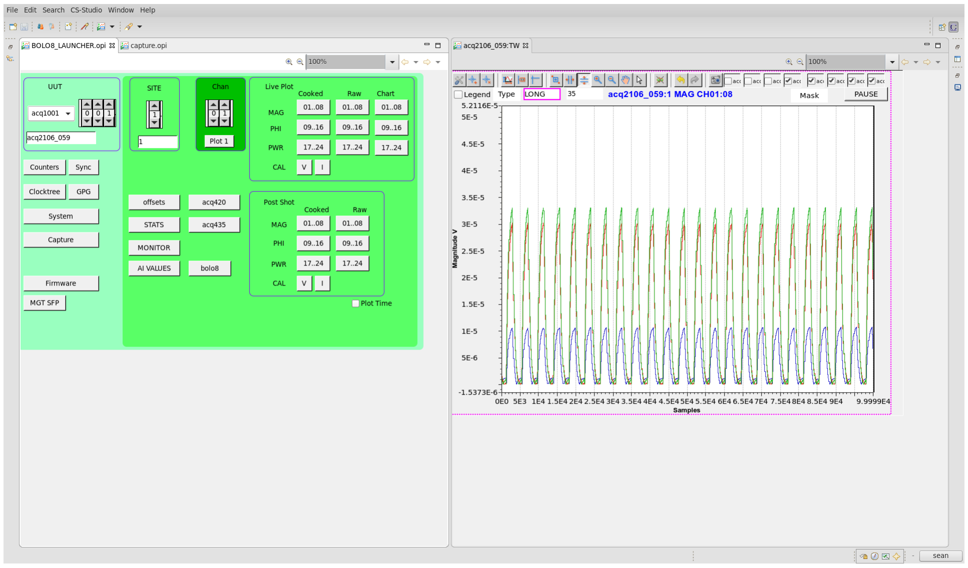
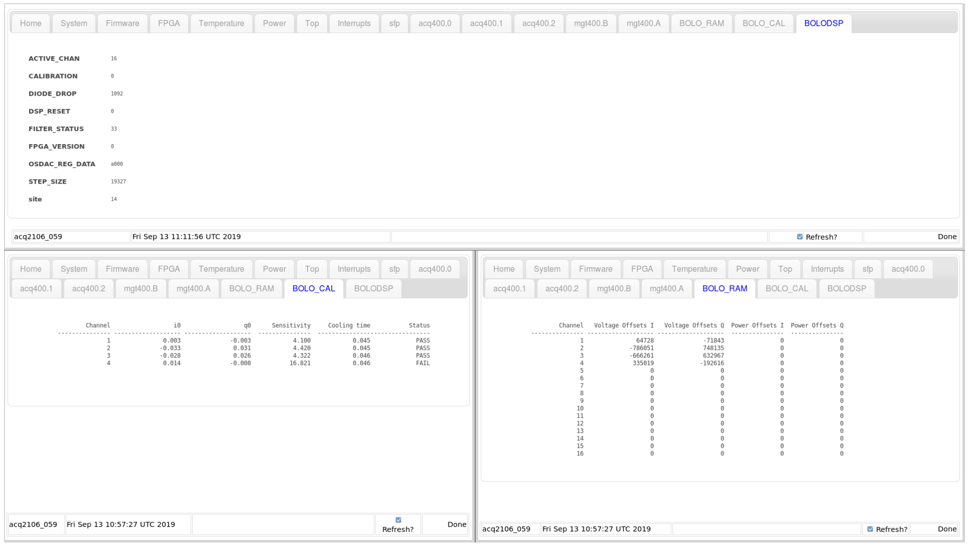


Figure 8: A CS Studio transient plot showing three working channels capturing light from a flashing bicycle lamp.

stream data from the acq2106 and use a flashing bicycle lamp to inspect the shark fin shape in the resultant data. Calibration removes the offsets observed in a non-calibrated system.

7 Appendix: Embedded web page diagnostics

It is also worth mentioning the embedded web page diagnostic tools which can be seen in figure 9. These tools allow the user to see the values that the last calibration returned, the calibration offsets that were loaded into the FPGA after the last calibration, and the DSP settings that are currently active.



ACTIVE_CHAN	16
CALIBRATION	0
DIODE_DROP	1092
DSP_RESET	0
FILTER_STATUS	33
FPGA_VERSION	0
OSDAC_REG_DATA	a000
STEP_SIZE	19327
site	14

acq2106_059 Fri Sep 13 11:11:56 UTC 2019

```
       Channel                i0                q0    Sensitivity    Cooling time            Status
-------------- ----------------- ----------------- -------------- ---------------------------------
             1             0.003            -0.003          4.100           0.045              PASS
             2            -0.033             0.031          4.420           0.045              PASS
             3            -0.028             0.026          4.322           0.046              PASS
             4             0.014            -0.008         16.821           0.046              FAIL
```

acq2106_059 Fri Sep 13 10:57:27 UTC 2019

```
       Channel  Voltage Offsets I  Voltage Offsets Q  Power Offsets I  Power Offsets Q
-------------- ------------------ ------------------ ---------------- ----------------
             1              64728             -71843                0                0
             2            -786051             748135                0                0
             3            -666261             632967                0                0
             4             335019            -192616                0                0
             5                  0                  0                0                0
             6                  0                  0                0                0
             7                  0                  0                0                0
             8                  0                  0                0                0
             9                  0                  0                0                0
            10                  0                  0                0                0
            11                  0                  0                0                0
            12                  0                  0                0                0
            13                  0                  0                0                0
            14                  0                  0                0                0
            15                  0                  0                0                0
            16                  0                  0                0                0
```

acq2106_059 Fri Sep 13 10:57:27 UTC 2019

Figure 9: Three firefox windows showing the BOLO embedded web pages.