

Status of the Optoboard microservice

19th March 2024

Daniele Dal Santo
on behalf of the Bern ATLAS Team

u^b

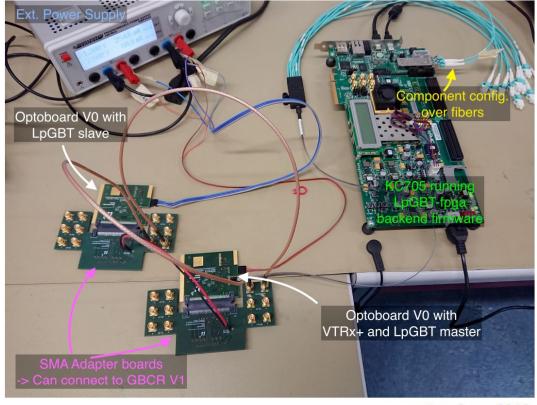
b
**UNIVERSITÄT
BERN**

AEC
ALBERT EINSTEIN CENTER
FOR FUNDAMENTAL PHYSICS

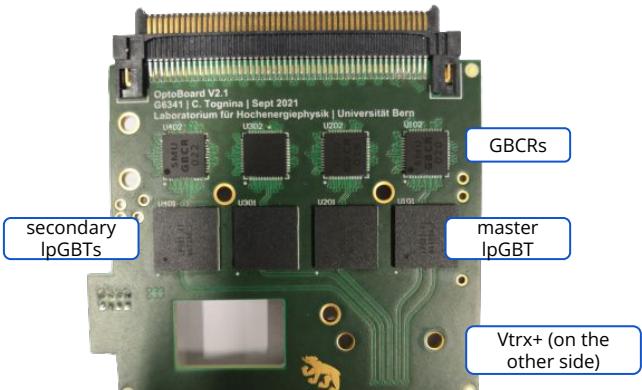


Recap of the Previous Update

1. Original software developed for configuring single IpGBT and VTRx+ on a Optoboard V0
 - a. no FELIX cards available at the time, KC705 FPGA used with custom firmware
 - b. supported communication tool: `fpgbtconf`
2. Optoboard V1: include slave ASICS → use IpGBT master I²C controller
 - a. expanded original scripts
3. Optoboard > V2.1: increased number of users
 optoboard_felix restructured with a more modular structure
 - a. introduction of *InitOpto* CLI entrypoint
 - b. added useful ASIC specific methods (see [here](#))
 - c. added support for IC communication over *felixcore* via *ic-over-netio*



setup from 2019



[Latest update \(01.02.23\)](#) on the Optoboard software by Roman Müller

optoboard-felix

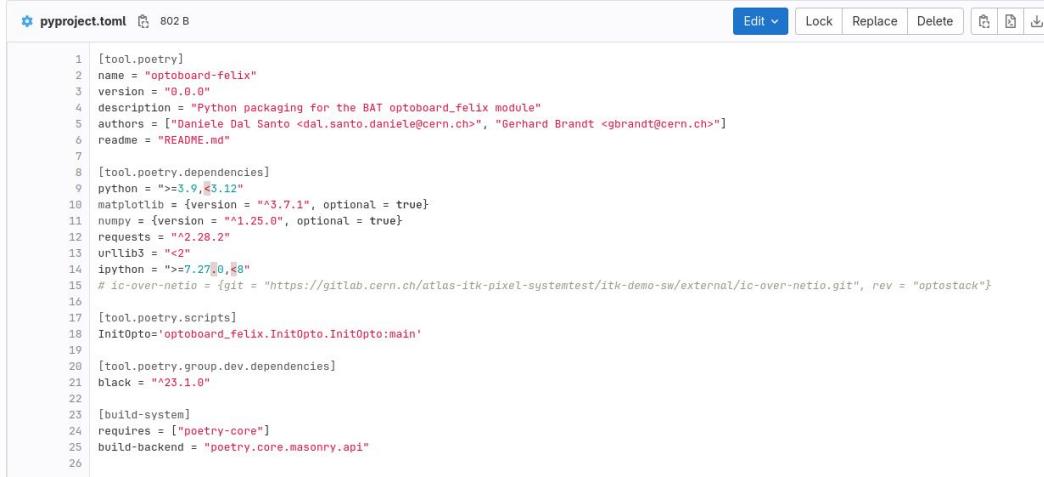
optoboard_felix reorganized as pip installable package

On repository tags:

- publish package to Package Registry
- automatic software documentation update

pip install optoboard-felix --index-url https://gitlab.cern.ch/api/v4/projects/113584/packages/pypi/simple

Latest tag: 1.0.34



The screenshot shows a GitHub code editor interface with the file 'pyproject.toml' open. The file contains Poetry configuration for a Python project named 'optoboard-felix'. The code includes dependencies like matplotlib, numpy, requests, urllib3, and ipython, and a build-system section with poetry-core and masonry-api requirements.

```
1 [tool.poetry]
2 name = "optoboard-felix"
3 version = "0.0.0"
4 description = "Python packaging for the BAT optoboard_felix module"
5 authors = ["Daniele Dal Santo <dal.santo.daniele@cern.ch>, Gerhard Brandt <gbrandt@cern.ch>"]
6 readme = "README.md"
7
8 [tool.poetry.dependencies]
9 python = ">=3.9,<3.12"
10 matplotlib = {version = "^3.7.1", optional = true}
11 numpy = {version = "1.25.0", optional = true}
12 requests = "2.28.2"
13 urllib3 = "c"
14 ipython = "7.27.0,<8"
15 # ic-over-netio = {git = "https://gitlab.cern.ch/atlas-itk-pixel-systemtest/itk-demo-sw/external/ic-over-netio.git", rev = "optostack"}
16
17 [tool.poetry.scripts]
18 InitOpto=optoboard_felix.InitOpto.InitOpto:main
19
20 [tool.poetry.group.dev.dependencies]
21 black = "^23.1.0"
22
23 [build-system]
24 requires = ["poetry-core"]
25 build-backend = "poetry.core.masonry.api"
26
```

optoboard-felix

InitOpto shim to run InitOpto.py and initialize optoboard instances

```
InitOpto [-h]
  [-P/-optoListPath OPTOLISTPATH]
  [-p/-config_path CONFIG_PATH]
  -s/-optoboard_serial OPTOBOARD_SERIAL
  [-v/-vtrx_v {1,2,1.3}]
  -G/-flx_G {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23}
  -d/-flx_d FLX_D
  [-T/-commToolName {"flpgbtconf", "flpgbtconfAPI", "ic-over-netio", "itk-ic-over-netio-next", "lpgbt-com"}]
  [-woflxcore {0,1}]
  [-c/-configure {0,1}]
  [-D/-debug {0,1}]
  [-t/-test_mode {0,1}]
  [-configInDB {0,1}]
  [--CONFIGDB_ADDRESS CONFIGDB_ADDRESS]
  [-i]
```

1 OB instance,
'opto'

```
InitOpto --optoboard_serial 4400072 --vtrx_v 1.3 --flx_G 0 --flx_d 0 -i --configure 1
>>> opto.lpgbt1.check_PUSM_status()
>>> opto.lpgbt1.swap_rx_polarity(0)
>>> opto.lpgbt1.swap_tx_polarity(0)
```

OB instances,
'optoList'

```
InitOpto --optoListPath optoConfigList_example.json -i
```

optoConfigList_example.json 459 B

```
1  {
2   "OB0": {
3     "optoboard_serial": "30000000",
4     "vtrx_v": "1.3",
5     "flx_G": 0,
6     "flx_d": 0,
7     "configure": 0,
8     "test_mode": 1,
9     "config_path": "",
10    "debug": 0,
11    "configInDB": 0,
12    "commToolName": ""
13  },
14  "OB1": {
15    "optoboard_serial": "33000000",
16    "vtrx_v": "1.3",
17    "flx_G": 0,
18    "flx_d": 0,
19    "configure": 0,
20    "test_mode": 1,
21    "config_path": "",
22    "debug": 0,
23    "configInDB": 0,
24    "commToolName": ""
25  }
26}
```

ConfigGUI.json has
this structure

optoboard-felix: InitOpto configuration

InitOpto requires:

- configuration of IC communication tool in [CommConfig.py](#)

```
TX_PORT = int(os.environ["TX_PORT"]) if "TX_PORT" in os.environ else 12340
RX_PORT = int(os.environ["RX_PORT"]) if "RX_PORT" in os.environ else 12350
TX_TAG = int(os.environ["TX_TAG"]) if "TX_TAG" in os.environ else 17
RX_TAG = int(os.environ["RX_TAG"]) if "RX_TAG" in os.environ else 29
INTERFACE = str(os.environ["INTERFACE"]) if "INTERFACE" in os.environ else "127.0.0.1"
FELIX_BUS = str(os.environ["FELIX_BUS"]) if "FELIX_BUS" in os.environ else "/bus"

CommConfig = { "I Chandler_config": { "ip": INTERFACE, "tx_port": TX_PORT, "rx_port": RX_PORT, "tx_tag": TX_TAG, "rx_tag": RX_TAG},  

    "lpgbt_to_felixStar_config": { "interface": INTERFACE, "bus_dir": FELIX_BUS, "tx_tag": TX_TAG, "rx_tag": RX_TAG}, }
```

both felixcore and
felixstar supported

- knowledge of composition of each Optoboard, given its serial number
 - [components.json](#) created from [productionDB](#)

```
"4200042": {
    "optoboard_v": 4,
    "lpgbt_v": 1,
    "lpgbt_master_addr": 116,
    "lpgbt1": 1,
    "lpgbt2": 0,
    "lpgbt3": 1,
    "lpgbt4": 0,
    "gbcr_v": 2,
    "gbcr1": 1,
    "gbcr2": 0,
    "gbcr3": 1,
    "gbcr4": 0,
    "efused": 0,
    "I2C_master": 0
},
```

optoboard-felix: optoboard configuration

Default configuration files are provided for all Optoboard versions, however the followings are setup specific:

- IGBT: rx/tx polarities & (phase)
- GBCR: equalization strength

Useful tools:

- soft error counter and scan
- parallel softerror counter
- (rx,tx) scan
- swap rx and tx polarity
- set gbcr equalization
- phase training
- phase BERT scan (soon available)
- ...

```
"Optoboard":  
{  
    "serial": "3200030",  
    "flx_6": 0,  
    "flx_d": 0,  
    "vtrx_v": "1.3"  
},
```

needs to be added to
custom config files

```
1 {  
2     "power_up_registers_master":  
3     {  
4         "CLKGCONF10",  
5         "CLKGCONF10",  
6         "CLKGPLINES*",  
7         "CLKGPLINTCUR",  
8         "CLKGPLP10POCUR",  
9         "CLKGCRPP10POCUR",  
10        "CLKGCRINTCUR",  
11        "CLKGCRPP10POCUR",  
12        "CLKGFLINTCUR",  
13        "CLKGFFCAP",  
14        "CLKGCTOVERRIDE1",  
15        "CLKGCTOVERRIDE1CAPBANK",  
16        "CLKGCTOVERRIDE1",  
17        "CLKGCTOVERRIDE1",  
18        "CLKL1COMP10",  
19        "EPRXOLCONFIN9",  
20        "EPRXOLCONFILT",  
21        "EPRXOLCONFILT",  
22        "EPRXON10",  
23        "EPRXOCONTROL",  
24        "EPRXOCNTN8",  
25        "EPRXICONTROL",  
26        "EPRXICNTN8",  
27        "EPRXICNTN8",  
28        "EPRXICNTN8",  
29        "EPRXICNTN8",  
30        "EPRXOCNTN8",  
31        "EPRXACONTROL",  
32        "EPRXACNTN8",  
33        "EPRXOCNTN8",  
34        "EPRXSCNTN7",  
35        "EQCONF16",  
36        "FAMAXHEADER10COUNT",  
37        "FAMAXHEADER10COUNTAFTERN",  
38        "FAMAXHEADER10COUNTAFTERN",  
39        "EPTXDATARATE",  
40        "EPTXENABLE",  
41        "EPTX32ENABLE",  
42        "EPTXOCNTN8",  
43        "EPTX101_00CNTNTR",  
44        "EPTX101_00CNTNTR",  
45        "EPTX121_20CNTNTR",  
46        "EPTX121_20CNTNTR",  
47        "EPTX111_10CNTNTR",  
48        "EPTX111_10CNTNTR",  
49        "EPTX111_12CNTNTR",  
50        "EPTX111_12CNTNTR",  
51        "EPTX211_20CNTNTR",  
52        "EPTX221_20CNTNTR",  
53        "EPTX223_22CNTNTR",  
54        "EPTX231_30CNTNTR",  
55        "EPTX233_30CNTNTR",  
56        "EPTX321_32CNTNTR",  
57        "EPTX331_32CNTNTR",  
58        "POWERUP2",  
59    },  
60    "power_up_registers_slave":  
61    {  
62        "CLKGBIASGENCONFIG": [8, 8, 8, 8],  
63        "CLKGCALIBRATIONENDOFCOUNT": [14, 14, 14, 14]  
64    },  
65    "CLKGCONFIG0":  
66    {  
67        "CDRCONTROLOVERRIDEENABLE": [0, 0, 0, 0],  
68        "CLKGCORES": [1, 1, 1, 1],  
69        "CLKGSABLEREFRAMEALIGNERLOCKCONTROL": [0, 0, 0, 0],  
70        "CLKGVCODAC": [8, 8, 8, 8],  
71        "CLKGVCORAILMODE": [1, 1, 1, 1]  
72    },  
73    "CLKGCONFIG1":  
74    {  
75        "CDRCONTROLOVERRIDEENABLE": [0, 0, 0, 0],  
76        "CLKGCORES": [1, 1, 1, 1],  
77        "CLKGSABLEREFRAMEALIGNERLOCKCONTROL": [0, 0, 0, 0],  
78        "CLKGVCODAC": [8, 8, 8, 8],  
79        "CLKGVCORAILMODE": [1, 1, 1, 1]  
80    },  
81    "CLKGFFCAP":  
82    {  
83        "CDRCOCONNECTCDR": [0, 0, 0, 0],  
84        "CLKGCBANKOVERRIDEENABLE": [0, 0, 0, 0],  
85        "CLKGFEEDFORWARDCAP": [3, 3, 3, 3],  
86        "CLKGFEEDFORWARDCAPWHENLOCKED": [3, 3, 3, 3]  
87    },  
88    "CLKGFLLINTCUR":  
89    {  
90        "CLKGFLLINTCUR": [5, 5, 5, 5],  
91        "CLKGFLLINTCURWHENLOCKED": [5, 5, 5, 5]  
92    },  
93    "CLKGLFCFG0":  
94    {  
95        "CLKGCBAPBANKSELECT": [0, 0, 0, 0],  
96        "CLKGLOCKFILTERCLKALWAYSON": [0, 0, 0, 0],  
97        "CLKGLOCKFILTERENABLE": [1, 1, 1, 1],  
98        "CLKGLOCKFILTERLOCKTHRCOUNTER": [15, 15, 15, 15]  
99    },  
100   "CLKGLFCFG1":  
101   {  
102       "CLKGLOCKFILTERRELLOCKTHRCOUNTER": [15, 15, 15, 15],  
103       "CLKGLOCKFILTERUNLOCKTHRCOUNTER": [15, 15, 15, 15]  
104   },  
105},  
106},  
107},  
108},  
109},  
110},  
111},  
112},  
113},  
114},  
115},  
116},  
117},  
118},  
119},  
120},  
121},  
122},  
123},  
124},  
125},  
126},  
127},  
128},  
129},  
130},  
131},  
132},  
133},  
134},  
135},  
136},  
137},  
138},  
139},  
140},  
141},  
142},  
143},  
144},  
145},  
146},  
147},  
148},  
149},  
150},  
151},  
152},  
153},  
154},  
155},  
156},  
157},  
158},  
159},  
160},  
161},  
162},  
163},  
164},  
165},  
166},  
167},  
168},  
169},  
170},  
171},  
172},  
173},  
174},  
175},  
176},  
177},  
178},  
179},  
180},  
181},  
182},  
183},  
184},  
185},  
186},  
187},  
188},  
189},  
190},  
191},  
192},  
193},  
194},  
195},  
196},  
197},  
198},  
199},  
200},  
201},  
202},  
203},  
204},  
205},  
206},  
207},  
208},  
209},  
210},  
211},  
212},  
213},  
214},  
215},  
216},  
217},  
218},  
219},  
220},  
221},  
222},  
223},  
224},  
225},  
226},  
227},  
228},  
229},  
230},  
231},  
232},  
233},  
234},  
235},  
236},  
237},  
238},  
239},  
240},  
241},  
242},  
243},  
244},  
245},  
246},  
247},  
248},  
249},  
250},  
251},  
252},  
253},  
254},  
255},  
256},  
257},  
258},  
259},  
260},  
261},  
262},  
263},  
264},  
265},  
266},  
267},  
268},  
269},  
270},  
271},  
272},  
273},  
274},  
275},  
276},  
277},  
278},  
279},  
280},  
281},  
282},  
283},  
284},  
285},  
286},  
287},  
288},  
289},  
290},  
291},  
292},  
293},  
294},  
295},  
296},  
297},  
298},  
299},  
300},  
301},  
302},  
303},  
304},  
305},  
306},  
307},  
308},  
309},  
310},  
311},  
312},  
313},  
314},  
315},  
316},  
317},  
318},  
319},  
320},  
321},  
322},  
323},  
324},  
325},  
326},  
327},  
328},  
329},  
330},  
331},  
332},  
333},  
334},  
335},  
336},  
337},  
338},  
339},  
340},  
341},  
342},  
343},  
344},  
345},  
346},  
347},  
348},  
349},  
350},  
351},  
352},  
353},  
354},  
355},  
356},  
357},  
358},  
359},  
360},  
361},  
362},  
363},  
364},  
365},  
366},  
367},  
368},  
369},  
370},  
371},  
372},  
373},  
374},  
375},  
376},  
377},  
378},  
379},  
380},  
381},  
382},  
383},  
384},  
385},  
386},  
387},  
388},  
389},  
390},  
391},  
392},  
393},  
394},  
395},  
396},  
397},  
398},  
399},  
400},  
401},  
402},  
403},  
404},  
405},  
406},  
407},  
408},  
409},  
410},  
411},  
412},  
413},  
414},  
415},  
416},  
417},  
418},  
419},  
420},  
421},  
422},  
423},  
424},  
425},  
426},  
427},  
428},  
429},  
430},  
431},  
432},  
433},  
434},  
435},  
436},  
437},  
438},  
439},  
440},  
441},  
442},  
443},  
444},  
445},  
446},  
447},  
448},  
449},  
450},  
451},  
452},  
453},  
454},  
455},  
456},  
457},  
458},  
459},  
460},  
461},  
462},  
463},  
464},  
465},  
466},  
467},  
468},  
469},  
470},  
471},  
472},  
473},  
474},  
475},  
476},  
477},  
478},  
479},  
480},  
481},  
482},  
483},  
484},  
485},  
486},  
487},  
488},  
489},  
490},  
491},  
492},  
493},  
494},  
495},  
496},  
497},  
498},  
499},  
500},  
501},  
502},  
503},  
504},  
505},  
506},  
507},  
508},  
509},  
510},  
511},  
512},  
513},  
514},  
515},  
516},  
517},  
518},  
519},  
520},  
521},  
522},  
523},  
524},  
525},  
526},  
527},  
528},  
529},  
530},  
531},  
532},  
533},  
534},  
535},  
536},  
537},  
538},  
539},  
540},  
541},  
542},  
543},  
544},  
545},  
546},  
547},  
548},  
549},  
550},  
551},  
552},  
553},  
554},  
555},  
556},  
557},  
558},  
559},  
560},  
561},  
562},  
563},  
564},  
565},  
566},  
567},  
568},  
569},  
570},  
571},  
572},  
573},  
574},  
575},  
576},  
577},  
578},  
579},  
580},  
581},  
582},  
583},  
584},  
585},  
586},  
587},  
588},  
589},  
590},  
591},  
592},  
593},  
594},  
595},  
596},  
597},  
598},  
599},  
600},  
601},  
602},  
603},  
604},  
605},  
606},  
607},  
608},  
609},  
610},  
611},  
612},  
613},  
614},  
615},  
616},  
617},  
618},  
619},  
620},  
621},  
622},  
623},  
624},  
625},  
626},  
627},  
628},  
629},  
630},  
631},  
632},  
633},  
634},  
635},  
636},  
637},  
638},  
639},  
640},  
641},  
642},  
643},  
644},  
645},  
646},  
647},  
648},  
649},  
650},  
651},  
652},  
653},  
654},  
655},  
656},  
657},  
658},  
659},  
660},  
661},  
662},  
663},  
664},  
665},  
666},  
667},  
668},  
669},  
670},  
671},  
672},  
673},  
674},  
675},  
676},  
677},  
678},  
679},  
680},  
681},  
682},  
683},  
684},  
685},  
686},  
687},  
688},  
689},  
690},  
691},  
692},  
693},  
694},  
695},  
696},  
697},  
698},  
699},  
700},  
701},  
702},  
703},  
704},  
705},  
706},  
707},  
708},  
709},  
710},  
711},  
712},  
713},  
714},  
715},  
716},  
717},  
718},  
719},  
720},  
721},  
722},  
723},  
724},  
725},  
726},  
727},  
728},  
729},  
730},  
731},  
732},  
733},  
734},  
735},  
736},  
737},  
738},  
739},  
740},  
741},  
742},  
743},  
744},  
745},  
746},  
747},  
748},  
749},  
750},  
751},  
752},  
753},  
754},  
755},  
756},  
757},  
758},  
759},  
760},  
761},  
762},  
763},  
764},  
765},  
766},  
767},  
768},  
769},  
770},  
771},  
772},  
773},  
774},  
775},  
776},  
777},  
778},  
779},  
780},  
781},  
782},  
783},  
784},  
785},  
786},  
787},  
788},  
789},  
790},  
791},  
792},  
793},  
794},  
795},  
796},  
797},  
798},  
799},  
800},  
801},  
802},  
803},  
804},  
805},  
806},  
807},  
808},  
809},  
810},  
811},  
812},  
813},  
814},  
815},  
816},  
817},  
818},  
819},  
820},  
821},  
822},  
823},  
824},  
825},  
826},  
827},  
828},  
829},  
830},  
831},  
832},  
833},  
834},  
835},  
836},  
837},  
838},  
839},  
840},  
841},  
842},  
843},  
844},  
845},  
846},  
847},  
848},  
849},  
850},  
851},  
852},  
853},  
854},  
855},  
856},  
857},  
858},  
859},  
860},  
861},  
862},  
863},  
864},  
865},  
866},  
867},  
868},  
869},  
870},  
871},  
872},  
873},  
874},  
875},  
876},  
877},  
878},  
879},  
880},  
881},  
882},  
883},  
884},  
885},  
886},  
887},  
888},  
889},  
890},  
891},  
892},  
893},  
894},  
895},  
896},  
897},  
898},  
899},  
900},  
901},  
902},  
903},  
904},  
905},  
906},  
907},  
908},  
909},  
910},  
911},  
912},  
913},  
914},  
915},  
916},  
917},  
918},  
919},  
920},  
921},  
922},  
923},  
924},  
925},  
926},  
927},  
928},  
929},  
930},  
931},  
932},  
933},  
934},  
935},  
936},  
937},  
938},  
939},  
940},  
941},  
942},  
943},  
944},  
945},  
946},  
947},  
948},  
949},  
950},  
951},  
952},  
953},  
954},  
955},  
956},  
957},  
958},  
959},  
960},  
961},  
962},  
963},  
964},  
965},  
966},  
967},  
968},  
969},  
970},  
971},  
972},  
973},  
974},  
975},  
976},  
977},  
978},  
979},  
980},  
981},  
982},  
983},  
984},  
985},  
986},  
987},  
988},  
989},  
990},  
991},  
992},  
993},  
994},  
995},  
996},  
997},  
998},  
999},  
1000},  
1001},  
1002},  
1003},  
1004},  
1005},  
1006},  
1007},  
1008},  
1009},  
1010},  
1011},  
1012},  
1013},  
1014},  
1015},  
1016},  
1017},  
1018},  
1019},  
1020},  
1021},  
1022},  
1023},  
1024},  
1025},  
1026},  
1027},  
1028},  
1029},  
1030},  
1031},  
1032},  
1033},  
1034},  
1035},  
1036},  
1037},  
1038},  
1039},  
1040},  
1041},  
1042},  
1043},  
1044},  
1045},  
1046},  
1047},  
1048},  
1049},  
1050},  
1051},  
1052},  
1053},  
1054},  
1055},  
1056},  
1057},  
1058},  
1059},  
1060},  
1061},  
1062},  
1063},  
1064},  
1065},  
1066},  
1067},  
1068},  
1069},  
1070},  
1071},  
1072},  
1073},  
1074},  
1075},  
1076},  
1077},  
1078},  
1079},  
1080},  
1081},  
1082},  
1083},  
1084},  
1085},  
1086},  
1087},  
1088},  
1089},  
1090},  
1091},  
1092},  
1093},  
1094},  
1095},  
1096},  
1097},  
1098},  
1099},  
1100},  
1101},  
1102},  
1103},  
1104},  
1105},  
1106},  
1107},  
1108},  
1109},  
1110},  
1111},  
1112},  
1113},  
1114},  
1115},  
1116},  
1117},  
1118},  
1119},  
1120},  
1121},  
1122},  
1123},  
1124},  
1125},  
1126},  
1127},  
1128},  
1129},  
1130},  
1131},  
1132},  
1133},  
1134},  
1135},  
1136},  
1137},  
1138},  
1139},  
1140},  
1141},  
1142},  
1143},  
1144},  
1145},  
1146},  
1147},  
1148},  
1149},  
1150},  
1151},  
1152},  
1153},  
1154},  
1155},  
1156},  
1157},  
1158},  
1159},  
1160},  
1161},  
1162},  
1163},  
1164},  
1165},  
1166},  
1167},  
1168},  
1169},  
1170},  
1171},  
1172},  
1173},  
1174},  
1175},  
1176},  
1177},  
1178},  
1179},  
1180},  
1181},  
1182},  
1183},  
1184},  
1185},  
1186},  
1187},  
1188},  
1189},  
1190},  
1191},  
1192},  
1193},  
1194},  
1195},  
1196},  
1197},  
1198},  
1199},  
1200},  
1201},  
1202},  
1203},  
1204},  
1205},  
1206},  
1207},  
1208},  
1209},  
1210},  
1211},  
1212},  
1213},  
1214},  
1215},  
1216},  
1217},  
1218},  
1219},  
1220},  
1221},  
1222},  
1223},  
1224},  
1225},  
1226},  
1227},  
1228},  
1229},  
1230},  
1231},  
1232},  
1233},  
1234},  
1235},  
1236},  
1237},  
1238},  
1239},  
1240},  
1241},  
1242},  
1243},  
1244},  
1245},  
1246},  
1247},  
1248},  
1249},  
1250},  
1251},  
1252},  
1253},  
1254},  
1255},  
1256},  
1257},  
1258},  
1259},  
1260},  
1261},  
1262},  
1263},  
1264},  
1265},  
1266},  
1267},  
1268},  
1269},  
1270},  
1271},  
1272},  
1273},  
1274},  
1275},  
1276},  
1277},  
1278},  
1279},  
1280},  
1281},  
1282},  
1283},  
1284},  
1285},  
1286},  
1287},  
1288},  
1289},  
1290},  
1291},  
1292},  
1293},  
1294},  
1295},  
1296},  
1297},  
1298},  
1299},  
1300},  
1301},  
1302},  
1303},  
1304},  
1305},  
1306},  
1307},  
1308},  
1309},  
1310},  
1311},  
1312},  
1313},  
1314},  
1315},  
1316},  
1317},  
1318},  
1319},  
1320},  
1321},  
1322},  
1323},  
1324},  
1325},  
1326},  
1327},  
1328},  
1329},  
1330},  
1331},  
1332},  
1333},  
1334},  
1335},  
1336},  
1337},  
1338},  
1339},  
1340},  
1341},  
1342},  
1343},  
1344},  
1345},  
1346},  
1347},  
1348},  
1349},  
1350},  
1351},  
1352},  
1353},  
1354},  
1355},  
1356},  
1357},  
1358},  
1359},  
1360},  
1361},  
1362},  
1363},  
1364},  
1365},  
1366},  
1367},  
1368},  
1369},  
1370},  
1371},  
1372},  
1373},  
1374},  
1375},  
1376},  
1377},  
1378},  
1379},  
1380},  
1381},  
1382},  
1383},  
1384},  
1385},  
1386},  
1387},  
1388},  
1389},  
1390},  
1391},  
1392},  
1393},  
1394},  
1395},  
1396},  
1397},  
1398},  
1399},  
1400},  
1401},  
1402},  
1403},  
1404},  
1405},  
1406},  
1407},  
1408},  
1409},  
1410},  
1411},  
1412},  
1413},  
1414},  
1415},  
1416},  
1417},  
1418},  
1419},  
1420},  
1421},  
1422},  
1423},  
1424},  
1425},  
1426},  
1427},  
1428},  
1429},  
1430},  
1431},  
1432},  
1433},  
1434},  
1435},  
1436},  
1437},  
1438},  
1439},  
1440},  
1441},  
1442},  
1443},  
1444},  
1445},  
1446},  
1447},  
1448},  
1449},  
1450},  
1451},  
1452},  
1453},  
1454},  
1455},  
1456},  
1457},  
1458},  
1459},  
1460},  
1461},  
1462},  
1463},  
1464},
```

optoboard-felix: outlook

Refine IC communication library for *felixStar*:

- move communication to the slaves ASICs to C++
- reorganize the read/write commands to target fields instead of registers

itk-demo-optoboard-api: image

Latest tag: *demi-2*

itk-demo-optoboard-api

- flask and gunicorn (API server)
- celery (distributed task queue)

opto-base-image: *almalinux:9-base*

- felix-05-00-03-rm5-stand-alone
- optoboard-felix
- ic-over-netio for *felixcore*
- lpgbt-com-next for *felixStar*

At container startup:

- clone user and group ID
- possible to start API server or celery WORKER based on env variable

optoboard-felix: usage

Usage of *optoboard-felix* (running Felix container is assumed):

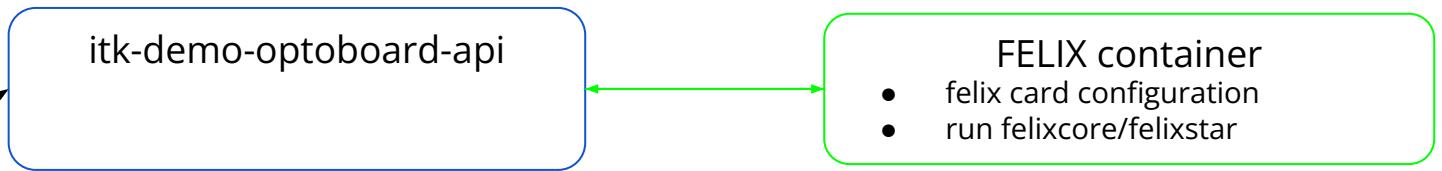
- **call *InitOpto*** in shell attached to *itk-demo-optoboard-api* container
- **call API endpoints** (requires API, WORKER & rabbitmq)
 - OB instances initialized based on *ConfigGUI.json*, mounted on worker
 - celery for task queuing and asynchronous execution → currently 1 queue per OB in an Optobox
 - individual queues controlled via flower
 - parallel OB configuration possible!
- **UI ...**

The screenshot shows a Swagger UI interface displaying a list of API endpoints defined in an *openapi.yaml* file. The endpoints are organized into sections: **default** and **OB**. The **default** section contains several GET and POST methods. The **OB** section contains a single POST method. A callout box at the bottom right points to the endpoint list, stating "endpoint list available [here](#) or on <http://localhost:5009/ui>".

Method	Path	Description
GET	/health	Check if optoboard service is unavailable
GET	/healthCelery	Check if optoboard service is unavailable
GET	/longTaskCelery	Check if optoboard service is unavailable
GET	/loadOptoList	Get json file and initialize optoboard list in case of changes to configuration file
GET	/getOptoGUIConf	Get the json file for the Configuration of the GUI
GET	/readLog	Read log file
default		
POST	/modifyOptoGUIConf	Get the json file for the Configuration of the GUI
POST	/initOptoList	Endpoint which defines the list of optoboards
POST	/statuscheck	Endpoint which provides information about the status of the optoboard
POST	/sendWriteRequest	Write register from name or hexadecimal address
POST	/sendReadRequest	Read register from name or hexadecimal address
POST	/configure	Does a configuration of the Optoboard based on a json file in server
POST	/configureAll	Does a configuration of the Optoboard based on a json file in server
POST	/BERT_wrapper	Does a BER Test
POST	/swapPolarity_wrapper	Swaps polarities of uplinks (tx)downlinks (rx)
POST	/setEqualizer_wrapper	Changes or monitors GBCR equalization settings
POST	/setLpGBTPhase_wrapper	Change phase mode of lpGBT

opto-stack: structure and entrypoints

**CLI
calls**



opto-stack: structure and entrypoints

**API
calls**

itk-demo-optoboard-api
API
• `http://$(hostname):5009`

rabbitmq

itk-demo-optoboard-api
WORKER
• (celery for asynchronous calls)

*ConfigGUI.json mounted
to /config*

**CLI
calls**

FELIX container
• felix card configuration
• run felixcore/felixstar

API usage

Custom tests can be easily designed by calling DCS, Felix, Optoboard, etc. APIs

```
def configureOB(OB: int):
    headers = {
        'accept': 'application/json',
        'Content-Type': 'application/json',
    }

    json_data = {
        'activeBcr': 'None',
        'activeLpGBT': 'None',
        'optoboardPosition': 'OB'+str(OB),
    }

    response = requests.post(OPTO_API+'/optoboard/api/configure', headers=headers, json=json_data)
    if response.status_code != 200:
        print("configureOB"+str(OB)+" failed")
    response_text = response.text.strip('\n')
    if response_text == "Optoboard configuration failed":
        print("Optoboard "+str(OB)+" configuration failed")
```

```
def getLinkAlignment(link = None):
    headers = {
        'accept': 'application/json',
    }

    response = requests.get(FELIX_API+'/getlink', headers=headers)
    if response.status_code != 200:
        print("getLinkAlignment failed")
    return

    linkalignment = response.json()
    if link:
        return linkalignment[str(link)]
    else:
        return linkalignment
```

L. Flores' script to test multiple OBs using API calls

itk-demo-optoboard-ui: image

- Image created in 2 steps:
 - *node:16.3.0* image to build JSX project
 - *nginx:stable-alpine* to serve the web content
- Web UI available on *http://\$(hostname --fqdn):5089*
- One panel per OB, based on *ConfigGUI.json* mounted to */config* in the worker

Optoboard GUI

Latest tag: *demi-2*

Optoboard GUI

Edit GUI Configuration

Select default configuration Choose Configuration Confirm

Basic myConfigName Confirm

```

1  "OB0": {
2    "ConfigurationOB0": "http://localhost:5000",
3    "connection": "ic-over-netio",
4    "configure": false,
5    "config_path": "",
6    "configure": 0,
7    "debug": false,
8    "flx_G": "0",
9    "flx_V": "0",
10   "optoboard_serial": "4400073",
11   "test_mode": false,
12   "vtrx_v": "1.3"
13 },
14 }
15 
```

- also possible to directly edit .json
- will be moved to configDB

```

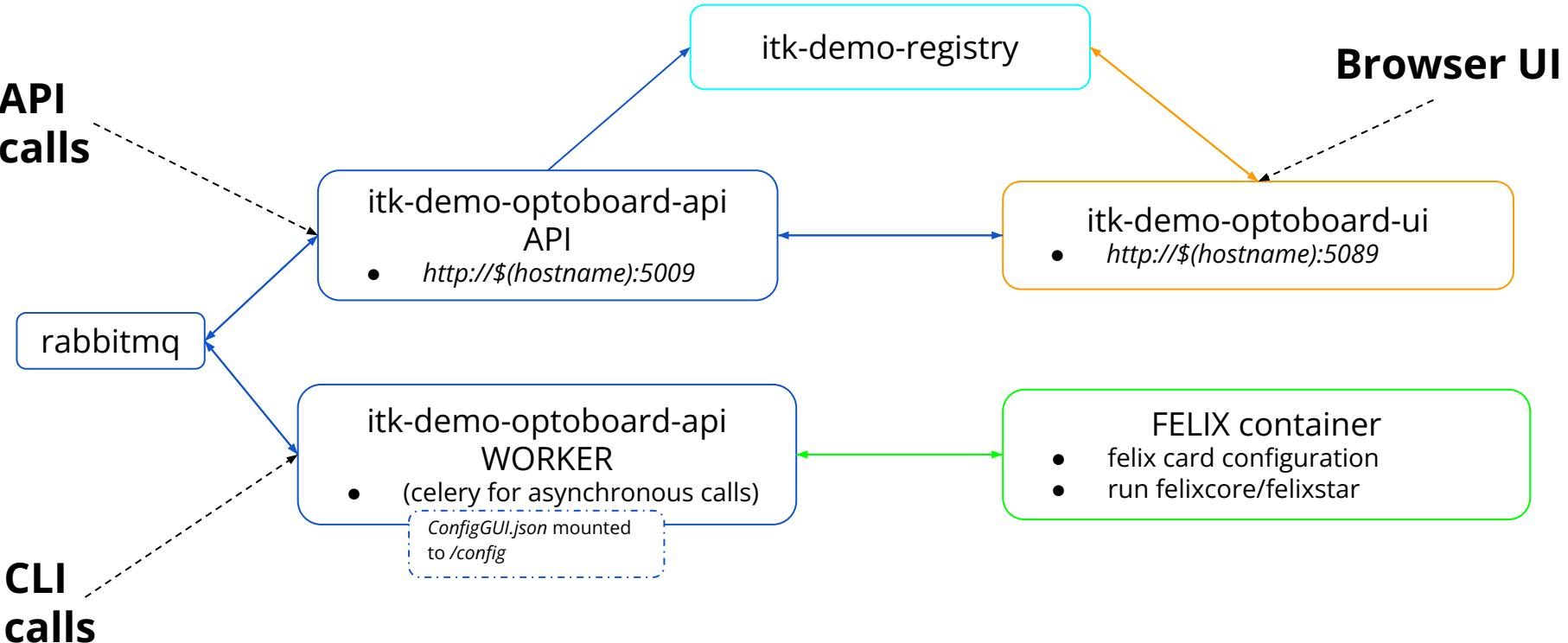
2023-11-29 16:17:11,083 - INFO - CommWrapper.py - Initialized IHandler with IP: 127.0.0.1, TX port: 12340, RX port: 12350, TX tag: 17, RX tag: 29
2023-11-29 16:17:11,084 - WARNING - CommWrapper.py - IHandler.connect() failed! Check if the method is available and if the IHandler configuration is correct!
2023-11-29 16:17:11,084 - INFO - CommWrapper.py - CommWrapper object initialised!
2023-11-29 16:17:11,085 - INFO - Optoboard.py - Logger level at INFO
2023-11-29 16:17:11,085 - INFO - Optoboard.py - component: OrderedDict([('optoboard_v', 3), ('lpGBT_v', 1), ('lpGBT_master_addr', 116), ('lpGBT1', 1), ('lpGBT2', 0), ('lpGBT3', 1), ('lpGBT4', 0), ('gbcr_v', 2), ('gbcr1', 1), ('gbcr2', 0), ('gbcr3', 0), ('gbcr4', 0), ('fused', 0), ('I2C_master', 0)])
2023-11-29 16:17:11,086 - INFO - LpGBT.py - lpGBT1 object initialised!
2023-11-29 16:17:11,086 - INFO - LpGBT.py - lpGBT3 object initialised!
2023-11-29 16:17:11,086 - INFO - Gbcr.py - gbcr1 object initialised!
2023-11-29 16:17:11,088 - INFO - Gbcr.py - gbcr3 object initialised!
2023-11-29 16:17:11,089 - INFO - Vtrx.py - vtrx object initialised!
2023-11-29 16:17:11,089 - INFO - Optoboard.py - Optoboard object initialised!
2023-11-29 16:17:11,089 - INFO - celeryTasks.py - Optoboard in position OB0 has been added!

```

Jump to the bottom

opto-stack: structure and entrypoints

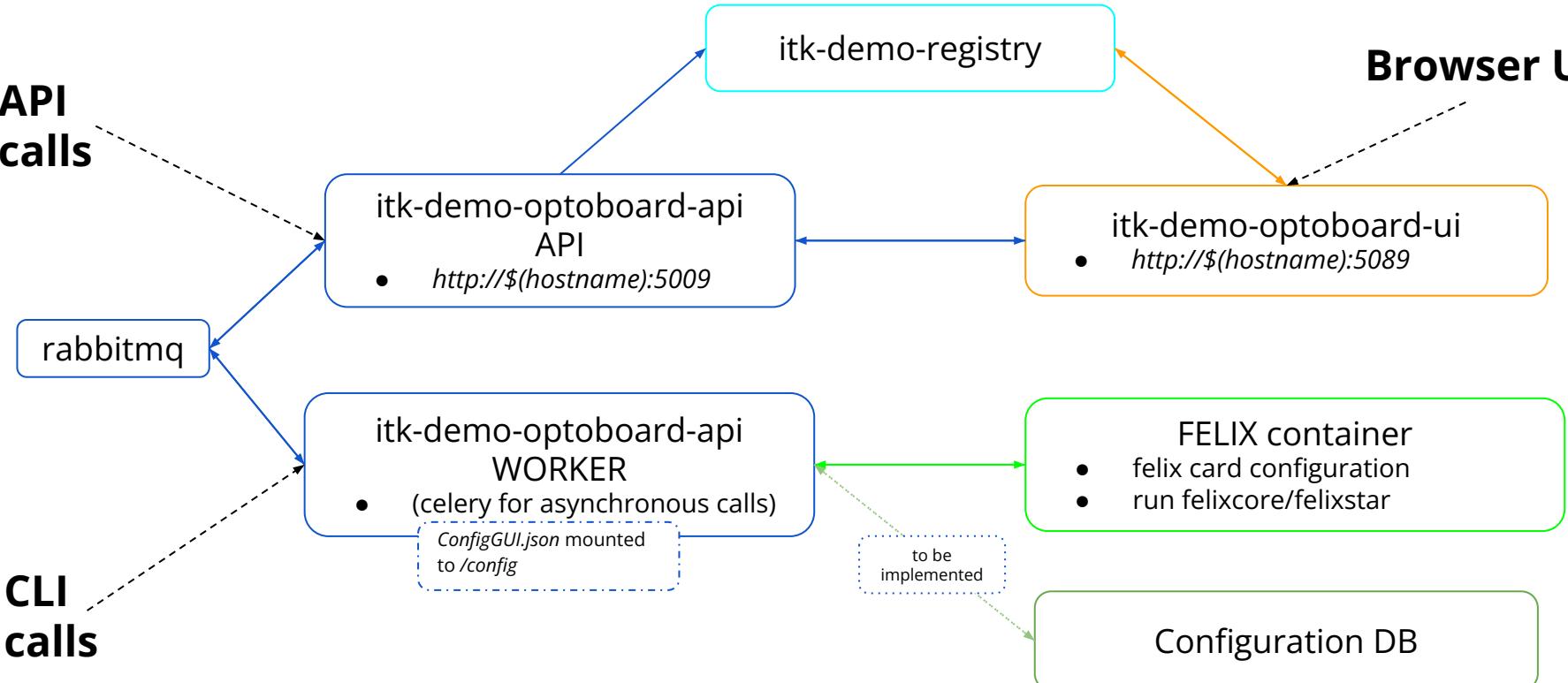
**API
calls**



opto-stack: structure and entrypoints

**API
calls**

Browser UI



Deployment, outlook and useful links

2 ways (follow the instructions in the READMEs):

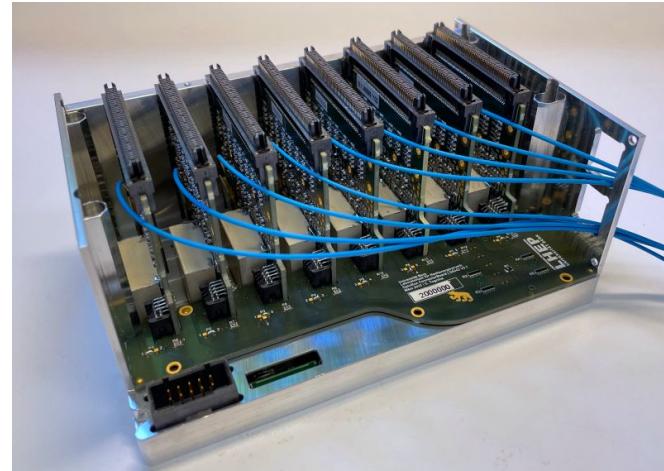
- via the stacks, using Docker, example [here](#)
- via the itk-demo-optoboard [example folder](#)

ITk-demo-optoboard outlook:

- restructure of celery backend
- improve error handling
- ...

Useful links:

- [Bern-Optoboard](#) mattermost channel
- [Optoboard software documentation](#)
- Report problems and requests
 - [optoboard felix issues](#)
 - [itk-demo-optoboard issues](#)



Status of the Optoboard microservice

19th March 2024

Daniele Dal Santo
on behalf of the Bern ATLAS Team

u^b

b
**UNIVERSITÄT
BERN**

AEC
ALBERT EINSTEIN CENTER
FOR FUNDAMENTAL PHYSICS

