**Blast to QuRT porting guide**

1. **Introduction**

BLAST has been RTOS for Hexagon processor (QDSP6). BLAST is renamed as QuRT to help our OpenDSP initiatives. QuRT stands for Qualcomm Real Time OS. As part of the name changing exercise, rules were established for QuRT APIs. Most of the APIs, types and constants except some of the Thread APIs and PIPE APIs can be made backward compatible. Released “blast.h” helps support backward compatibility.

The Blast directory is replaced by Qurt directory.

To provide the backward compatible for name changed, the blast.h header file supports the Blast to QuRT backward compatible for

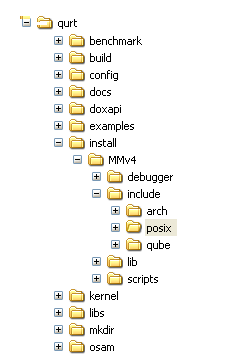
* Api funtions names changes
* Constants and macros changes
* Typedefs and structure names changes
* Shared global variables names changes

For Blast to QuRT porting, user needs to include “blast.h” when using the blast API functions.

Please refer to Appendix B for Lpass Porting example.

This document provides the mapping tables of API functions, structure, typedef, constant, and macros from Blast to QuRt.

The QuRT directory structure



**Pager Blast VU dependency**

For the targets which support Blast demand paging – dl\_pager\_blast

P4 path //source/qcom/qct/core/kernel/dl\_pager\_blast/ .

Demand pager VU associates with QuRT

**VU\_CORE\_KERNEL\_PAGER\_BLAST.00.00.06**

1. **Non backward compatible APIs**

For those QuRT API functions which have name and parameters changed, the QuRT API wrapper functions are provided to make them backward compatible to blast API functions

*Table 1: Non backward compatible API functions mappings*

|  |
| --- |
| **Blast API function Names** |
| blast\_thread\_create |
| blast\_thread\_set\_name |
| blast\_thread\_set\_name |
| blast\_yield |
| blast\_thread\_setattr |
| blast\_thread\_set\_prefetch |
| blast\_thread\_set\_gp |
| blast\_thread\_set\_ugp |
| blast\_pipe\_free |
| blast\_pipe\_trysend |
| blast\_pipe\_tryrecv |
| blast\_pipe\_create |
| blast\_pipe\_alloc |

* 1. **Thread APIs**

The Qurt thread APIs functions are changed. They are not compatible with the Blast thread API.

To use QuRT thread APIs, the qurt\_thread\_attr\_init([qurt\_thread\_attr\_t](file:///Z:\opham\workspace\blastrel241latest\qurt\main\latest\doxapi\output\html\struct__qurt__thread__attr.html) \*thread\_attr) need to call to setup the default thread attributes

qurt\_thread\_attr\_init([qurt\_thread\_attr\_t](file:///Z:\opham\workspace\blastrel241latest\qurt\main\latest\doxapi\output\html\struct__qurt__thread__attr.html) \*thread\_attr) initializes the thread attributes to default values

* QURT\_THREAD\_ATTR\_NAME\_MAXLEN 16
* QURT\_THREAD\_ATTR\_TCB\_PARTITION\_DEFAULT 0
* QURT\_THREAD\_ATTR\_PRIORITY\_DEFAULT 256
* QURT\_THREAD\_ATTR\_ASID\_DEFAULT 0
* QURT\_THREAD\_ATTR\_AFFINITY\_DEFAULT (-1)
* QURT\_THREAD\_ATTR\_TIMETEST\_ID\_DEFAULT (-2)

To set a specific attributes for a thread using these following api functions

static inline void qurt\_thread\_attr\_set\_name (qurt\_thread\_attr\_t \*attr, char \*name)

static inline void qurt\_thread\_attr\_set\_tcb\_partition (qurt\_thread\_attr\_t \*attr, unsigned short tcb\_partition)

static inline void qurt\_thread\_attr\_set\_priority (qurt\_thread\_attr\_t \*attr, unsigned short priority)

static inline void qurt\_thread\_attr\_set\_affinity (qurt\_thread\_attr\_t \*attr, unsigned char affinity)

static inline void qurt\_thread\_attr\_set\_timetest\_id (qurt\_thread\_attr\_t \*attr, unsigned short timetest\_id)

static inline void qurt\_thread\_attr\_set\_stack\_size (qurt\_thread\_attr\_t \*attr, unsigned int stack\_size)

static inline void qurt\_thread\_attr\_set\_stack\_addr (qurt\_thread\_attr\_t \*attr, void \*stack\_addr)

* + 1. **Sample of how to replace blast\_thread\_create by qurt\_thread\_create**

|  |  |  |  |
| --- | --- | --- | --- |
| iint **qurt\_thread\_create** | ( | [qurt\_thread\_t](file:///Z:\opham\workspace\blastrel241latest\qurt\main\latest\doxapi\output\html\qurt__thread_8h.html#ee47afd498b75df65c089f744befb2e8) \* | **thread\_id***,* |
|  |  | [qurt\_thread\_attr\_t](file:///Z:\opham\workspace\blastrel241latest\qurt\main\latest\doxapi\output\html\struct__qurt__thread__attr.html) \* | **attr***,* |
|  |  | void(\*)(void \*) | **Entrypoint** |
|  |  | void \* | **arg** |  |
|  | ) |  |  |  |

**Parameters:**

|  |  |  |
| --- | --- | --- |
|  | attr | pointer to the thread attributes such as stack addr |
|  | entrypoint | Entry function of the thread |
|  | arg | Argument |

*thread\_id* Thread ID

**Returns:**

Qurt thread ID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| int **blast\_thread\_create** | ( | void \* | **pc***,* | |
|  |  | void \* | **stack***,* | |
|  |  | void \* | **arg***,* | |
|  |  | unsigned int | **prio***,* | |
|  |  | unsigned int | **stack\_size***,* | |
|  |  | unsigned int | **thread\_cfg** | |  | |
|  | **)** |  |  |  | |

**Parameters:**

|  |  |  |
| --- | --- | --- |
|  | pc | start PC |
|  | stack | Stack pointer |
|  | arg | Argument |
|  | stack\_size | Stack size |
|  | asid | ASID(Space ID) |
|  | thread\_cfg | The thread configuration field. The 8bits of LSB is hw\_bitmask that the thread will be restricted to run on. Bit 0 is corresponding to hw thread 0, and so on. If the corresponding bit is set to 1, then the software thread is eligible to run this hw thread. 0x3f means it can run any hw threads 0x0 also means it can run on any hw threads |

**Returns:**

Blast thread ID

Example:

Void threadMain(void)

{

…

}

int sample\_qurt\_thread\_create (void)

{

#define STACK\_SIZE 2048

void \*stack[STACK\_SIZE];

static char tname[16] = “TEST”

qurt\_thread\_attr\_t thread\_attr;

qurt\_thread\_t thread\_id;

int prio = 100;

int ret;

void \*arg =NULL;

qurt\_thread\_attr\_init (&thread\_attr);

qurt\_thread\_attr\_set\_name(&thread\_attr, (char\*)tname);

qurt\_thread\_attr\_set\_stack\_size (&thread\_attr, STACK\_SIZE);

qurt\_thread\_attr\_set\_stack\_addr (&thread\_attr, stack);

qurt\_thread\_attr\_set\_priority (&thread\_attr, (unsigned short)prio);

ret = qurt\_thread\_create (&thread\_id, &thread\_attr, (void \*)threadMain, arg);

if (ret == QURT\_EOK) {

ret = thread\_id;

}

return ret;

}

* + 1. **Sample of Thread set name**

The Blast\_thread\_set\_name function and qurt\_thread\_set\_name function are not compatible. Since the function’s parameters are different, the qurt\_thread\_set\_name\_wrapper function was created to provide the backward compatibility.

|  |  |  |  |
| --- | --- | --- | --- |
| **void blast\_thread\_set\_name** | ( | unsigned long long | *name0,* |
|  |  | unsigned long long | *name1* |  |
|  | ) |  |  |  |

**Parameters**:

|  |  |  |
| --- | --- | --- |
|  | *name0* | The first 8 bytes |
|  | *name1* | The second 8 bytes |
|  |  |  |
|  |  |  |

void sample\_qurt\_thread\_set\_name(unsigned long long name0, unsigned long long name1)

{

struct QURT\_ugp\_ptr \*pUgp;

char cname[16];

int lsize;

lsize = sizeof(unsigned long long);

memcpy(&cname[0],(char \*)&name0, lsize);

memcpy(&cname[lsize],(char \*)&name1, lsize);

qurt\_get\_my\_utcb(pUgp);

qurt\_thread\_attr\_set\_name (pUgp->utcb.attr.name , cname );

}

* 1. **Pipe APIs**
* New QuRT pipe apis for pipe attribute structure

typedef struct {

qurt\_pipe\_data\_t \*buffer;

unsigned int elements;

} qurt\_pipe\_attr\_t

static inline void qurt\_pipe\_attr\_set\_buffer(qurt\_pipe\_attr\_t \*attr, qurt\_pipe\_data\_t \*buffer)

static inline void qurt\_pipe\_attr\_set\_elements(qurt\_pipe\_attr\_t \*attr, unsigned int elements)

* Modified and non-backward compatible QuRT pipe APIs

int qurt\_pipe\_create (qurt\_pipe\_t \*\*pipe, qurt\_pipe\_attr\_t \*attr);

int qurt\_pipe\_init (qurt\_pipe\_t \*pipe, qurt\_pipe\_attr\_t \*attr);

int qurt\_pipe\_delete (qurt\_pipe\_t \*pipe)

* The difference between “init” and “create” in QuRT is “create” allocates the data structure, where as init takes the allocated data structure.
* This helps us expand mutex/pipe etc. to multi-PD.
* Some of the services may not have both “init” and “create” functionality. Since memory is allocated as part of the create API, it needs extra “\*” compared to “init”.
  1. **Power collapse APIs**

Qurt\_power\_shutdown\_enter\_wrapper with parameter L2 cache retention mode set to QURT\_POWER\_SHUTDOWN\_TYPE\_L2NORET. It provides a backward compatible with

blast\_ power\_shutdown\_enter.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| int qurt\_power\_shutdown\_enter | ( | int | *type* | ) |  |

Trigger power collapse. Determine if no valid interrupts are pending. (Valid interrupts are determined using the interrupt registration status.) If so, put the current hardware thread into WAIT mode. When all hardware threads have entered WAIT mode, the Hexagon processor can enter power-collapse mode.

**Parameters:**

|  |  |  |
| --- | --- | --- |
|  | *type* | The L2 cache retention type  QURT\_POWER\_SHUTDOWN\_TYPE\_L2NORET  QURT\_POWER\_SHUTDOWN\_TYPE\_L2RET |

**Returns:**

EOK: Operation successfully performed NOTE - This value indicates that the processor has returned from power-collapse mode.

non-zero: IPEND mask value indicating pending interrupts.

int qurt\_power\_shutdown\_enter\_wrapper (void)

{

int rc;

rc = qurt\_power\_shutdown\_enter(QURT\_POWER\_SHUTDOWN\_TYPE\_L2NORET );

return rc;

}

1. **APIs with backward compatibility**

This session has the API function names change only. There is no parameter changes.

Two API function mapping tables are

* Mapping Blast function name to Qurt function name
* Mapping Qmem function nane to Qurt function
  1. **API with just blast to qurt name changed only**

*Table 1:Blast to QuRT API function name changed only mappings*

|  |  |
| --- | --- |
| **Blast API function Names** | **QuRT API function Names** |
| **signal functions** | |
| blast\_allsignal\_get | qurt\_allsignal\_get |
| blast\_allsignal | qurt\_allsignal |
| blast\_allsignal\_destroy | qurt\_allsignal\_destroy |
| blast\_allsignal\_wait | qurt\_allsignal\_wait |
| blast\_allsignal\_init | qurt\_allsignal\_init |
| blast\_allsignal\_signal | qurt\_allsignal\_set |
| blast\_anysignal | qurt\_anysignal |
| blast\_anysignal\_init | qurt\_anysignal\_init |
| blast\_anysignal\_destroy | qurt\_anysignal\_destroy |
| blast\_anysignal\_clear | qurt\_anysignal\_clear |
| blast\_anysignal\_wait | qurt\_anysignal\_wait |
| blast\_anysignal\_set | qurt\_anysignal\_set |
| blast\_anysignal\_get | qurt\_anysignal\_get |
| **Atomic 32 bits functions** | |
| blast\_atomic\_add | qurt\_atomic\_add |
| blast\_atomic\_add\_return | qurt\_atomic\_add\_return |
| blast\_atomic\_and | qurt\_atomic\_and |
| blast\_atomic\_and\_return | qurt\_atomic\_and\_return |
| blast\_atomic\_barrier | qurt\_atomic\_barrier |
| blast\_atomic\_barrier\_smp | qurt\_atomic\_barrier\_smp |
| blast\_atomic\_barrier\_read | qurt\_atomic\_barrier\_read |
| blast\_atomic\_barrier\_read\_smp | qurt\_atomic\_barrier\_read\_smp |
| blast\_atomic\_barrier\_write | qurt\_atomic\_barrier\_write |
| blast\_atomic\_barrier\_write\_smp | qurt\_atomic\_barrier\_write\_smp |
| blast\_atomic\_clear\_bit | qurt\_atomic\_clear\_bit |
| blast\_atomic\_compare\_and\_set | qurt\_atomic\_compare\_and\_set |
| blast\_atomic\_change\_bit | qurt\_atomic\_change\_bit |
| blast\_atomic\_dec | qurt\_atomic\_dec |
| blast\_atomic\_dec\_return | qurt\_atomic\_dec\_return |
| blast\_atomic\_inc | qurt\_atomic\_inc |
| blast\_atomic\_inc\_return | qurt\_atomic\_inc\_return |
| blast\_atomic\_or | qurt\_atomic\_or |
| blast\_atomic\_or\_return | qurt\_atomic\_or\_return |
| blast\_atomic\_set | qurt\_atomic\_set |
| blast\_atomic\_set\_bit | qurt\_atomic\_set\_bit |
| blast\_atomic\_sub | qurt\_atomic\_sub |
| blast\_atomic\_sub\_return | qurt\_atomic\_sub\_return |
| blast\_atomic\_xor | qurt\_atomic\_xor |
| blast\_atomic\_xor\_return | qurt\_atomic\_xor\_return |
| **Atomic 64 bits functions** | |
| blast\_atomic64\_add | qurt\_atomic64\_add |
| blast\_atomic64\_add\_return | qurt\_atomic64\_add\_return |
| blast\_atomic64\_and | qurt\_atomic64\_and |
| blast\_atomic64\_and\_return | qurt\_atomic64\_and\_return |
| blast\_atomic64\_barrier | qurt\_atomic64\_barrier |
| blast\_atomic64\_barrier\_smp | qurt\_atomic64\_barrier\_smp |
| blast\_atomic64\_barrier\_read | qurt\_atomic64\_barrier\_read |
| blast\_atomic64\_barrier\_read\_smp | qurt\_atomic64\_barrier\_read\_smp |
| blast\_atomic64\_barrier\_write | qurt\_atomic64\_barrier\_write |
| blast\_atomic64\_barrier\_write\_smp | qurt\_atomic64\_barrier\_write\_smp |
| blast\_atomic64\_clear\_bit | qurt\_atomic64\_clear\_bit |
| blast\_atomic64\_compare\_and\_set | qurt\_atomic64\_compare\_and\_set |
| blast\_atomic64\_change\_bit | qurt\_atomic64\_change\_bit |
| blast\_atomic64\_dec | qurt\_atomic64\_dec |
| blast\_atomic64\_dec\_return | qurt\_atomic64\_dec\_return |
| blast\_atomic64\_inc | qurt\_atomic64\_inc |
| blast\_atomic64\_inc\_return | qurt\_atomic64\_inc\_return |
| blast\_atomic64\_or | qurt\_atomic64\_or |
| blast\_atomic64\_or\_return | qurt\_atomic64\_or\_return |
| blast\_atomic64\_set | qurt\_atomic64\_set |
| blast\_atomic64\_set\_bit | qurt\_atomic64\_set\_bit |
| blast\_atomic64\_sub | qurt\_atomic64\_sub |
| blast\_atomic64\_sub\_return | qurt\_atomic64\_sub\_return |
| blast\_atomic64\_xor | qurt\_atomic64\_xor |
| blast\_atomic64\_xor\_return | qurt\_atomic64\_xor\_return |
| **Barrier functions** | |
| blast\_barrier\_wait | qurt\_barrier\_wait |
| blast\_barrier | qurt\_barrier |
| blast\_barrier\_destroy | qurt\_barrier\_destroy |
| blast\_barrier\_init | qurt\_barrier\_init |
| **Cond functions** | |
| blast\_cond\_init | qurt\_cond\_init |
| blast\_cond\_destroy | qurt\_cond\_destroy |
| blast\_cond\_signal | qurt\_cond\_signal |
| blast\_cond\_wait | qurt\_cond\_wait |
| blast\_cond\_broadcast | qurt\_cond\_broadcast |
| **Profile pcycle/tcycle functions** | |
| blast\_get\_core\_pcycles | qurt\_get\_core\_pcycles |
| blast\_get\_pcycles | qurt\_get\_pcycles |
| blast\_get\_tcycles | qurt\_get\_tcycles |
| blast\_profile\_reset\_idle\_pcycles | qurt\_profile\_reset\_idle\_pcycles |
| blast\_profile\_get\_thread\_pcycles | qurt\_profile\_get\_thread\_pcycles |
| blast\_profile\_get\_idle\_pcycles | qurt\_profile\_get\_idle\_pcycles |
| blast\_profile\_enable | qurt\_profile\_enable |
| blast\_profile\_reset\_thread\_pcycles | qurt\_profile\_reset\_thread\_pcycles |
|  | |
| blast\_arch\_version | qurt\_arch\_version |
| blast\_receive\_system\_event | qurt\_receive\_system\_event |
| blast\_get\_env\_object | qurt\_get\_env\_object |
| **Interrupt - fastint functions** | |
| blast\_isr\_deregister | qurt\_isr\_deregister |
| blast\_isr\_register | qurt\_isr\_register |
| blast\_interrupt\_status | qurt\_interrupt\_status |
| blast\_interrupt\_clear | qurt\_interrupt\_clear |
| **Lifo functions** | |
| blast\_lifo\_push | qurt\_lifo\_push |
| blast\_lifo\_pop | qurt\_lifo\_pop |
| **Memory functions** | |
| blast\_realloc | qurt\_realloc |
| blast\_malloc | qurt\_malloc |
| blast\_free | qurt\_free |
| blast\_calloc | qurt\_calloc |
| blast\_mem\_map\_static\_query | qurt\_mem\_map\_static\_query |
| blast\_mem\_pool\_attach | qurt\_mem\_pool\_attach |
| blast\_mem\_internal | qurt\_mem\_internal |
| blast\_mem\_region\_create | qurt\_mem\_region\_create |
| blast\_mem\_region\_delete | qurt\_mem\_region\_delete |
| blast\_mem\_region\_query | qurt\_mem\_region\_query |
| blast\_mem\_region\_attr\_init | qurt\_mem\_region\_attr\_init |
| blast\_lookup\_physaddr | qurt\_lookup\_physaddr |
| **Cache functions** | |
| blast\_mem\_kernel\_cache\_opt | qurt\_mem\_kernel\_cache\_opt |
| blast\_mem\_kernel\_cache\_clean | qurt\_mem\_kernel\_cache\_clean |
| blast\_mem\_cache\_clean | qurt\_mem\_cache\_clean |
| blast\_mem\_cache\_flushall | qurt\_mem\_cache\_flushall |
| blast\_mem\_cache\_clean\_aligning | qurt\_mem\_cache\_clean\_aligning |
| **Pipe functions** | |
| blast\_pipe\_send | qurt\_pipe\_send |
| blast\_pipe\_recv | qurt\_pipe\_receive |
| **PMU functions** | |
| blast\_pmu\_get | qurt\_pmu\_get |
| blast\_pmu\_set | qurt\_pmu\_set |
| blast\_pmu\_enable | qurt\_pmu\_enable |
| **Power control functions** | |
| blast\_power\_control | qurt\_power\_control |
| blast\_power\_shutdown\_prepare | qurt\_power\_shutdown\_prepare |
| blast\_power\_shutdown\_exit | qurt\_power\_shutdown\_exit |
| blast\_power\_shutdown\_fail\_exit | qurt\_power\_shutdown\_fail\_exit |
| blast\_power\_tcxo\_prepare | qurt\_power\_tcxo\_prepare |
| blast\_power\_tcxo\_enter | qurt\_power\_tcxo\_enter |
| blast\_power\_tcxo\_exit | qurt\_power\_tcxo\_exit |
| blast\_power\_tcxo\_fail\_exit | qurt\_power\_tcxo\_fail\_exit |
| blast\_power\_wait\_for\_idle | qurt\_power\_wait\_for\_idle |
| blast\_power\_wait\_for\_active | qurt\_power\_wait\_for\_active |
| blast\_system\_ipend\_get | qurt\_system\_ipend\_get |
| blast\_system\_ipend\_clear | qurt\_system\_ipend\_clear |
| **Futex functions** | |
| blast\_futex\_wait | qurt\_futex\_wait |
| blast\_futex\_wake | qurt\_futex\_wake |
| blast\_futex\_free\_wait\_queue | qurt\_futex\_free\_wait\_queue |
| blast\_futex\_alloc\_wait\_queue | qurt\_futex\_alloc\_wait\_queue |
| **Mutex functions** | |
| blast\_rmutex\_lock | qurt\_rmutex\_lock |
| blast\_rmutex\_unlock | qurt\_rmutex\_unlock |
| blast\_rmutex\_destroy | qurt\_rmutex\_destroy |
| blast\_rmutex\_init | qurt\_rmutex\_init |
| blast\_mutex\_lock | qurt\_mutex\_lock |
| blast\_mutex\_init | qurt\_mutex\_init |
| blast\_mutex\_destroy | qurt\_mutex\_destroy |
| blast\_mutex\_unlock | qurt\_mutex\_unlock |
| **Pimutex functions** | |
| blast\_pimutex\_destroy | qurt\_pimutex\_destroy |
| blast\_pimutex\_unlock | qurt\_pimutex\_unlock |
| blast\_pimutex | qurt\_pimutex |
| blast\_pimutex\_lock | qurt\_pimutex\_lock |
| blast\_pimutex\_init | qurt\_pimutex\_init |
| **Sem functions** | |
| blast\_sem\_init | qurt\_sem\_init |
| blast\_sem\_add | qurt\_sem\_add |
| blast\_sem\_init\_val | qurt\_sem\_init\_val |
| blast\_sem\_get\_val | qurt\_sem\_get\_val |
| blast\_sem\_up | qurt\_sem\_up |
| blast\_sem\_destroy | qurt\_sem\_destroy |
| blast\_sem\_down | qurt\_sem\_down |
| **Thread functions** | |
| blast\_thread\_wait\_for\_active | qurt\_power\_wait\_for\_active |
| blast\_thread\_wait\_for\_idle | qurt\_power\_wait\_for\_idle |
| blast\_thread\_suspend | qurt\_thread\_suspend |
| blast\_thread\_resume | qurt\_thread\_resume |
| blast\_thread\_join | qurt\_thread\_join |
| blast\_thread\_stop | qurt\_thread\_stop |
| blast\_thread\_exit | qurt\_thread\_exit |
| blast\_thread\_set\_prefetch | qurt\_thread\_set\_prefetch |
| blast\_thread\_create | qurt\_thread\_create |
| blast\_thread\_set\_name | qurt\_thread\_set\_name |
| **Tlb funtions** | |
| blast\_tlb\_remove | qurt\_tlb\_remove |
| blast\_tls\_init | qurt\_tls\_init |
| BLAST\_tls\_reserve | QURT\_tls\_reserve |
| **Trace funtions** | |
| blast\_get\_trace\_marker | qurt\_trace\_get\_marker |
| blast\_has\_preempt\_trace | qurt\_trace\_changed |
| blast\_printf | qurt\_printf |
| blast\_yield | qurt\_yield |
| **Timer funtions** | |
| blast\_timer\_create | qurt\_timer\_create |
| blast\_timer\_delete | qurt\_timer\_delete |
| blast\_timer\_stop | qurt\_timer\_stop |
| blast\_timer\_restart | qurt\_timer\_restart |
| blast\_timer\_sleep | qurt\_timer\_sleep |

* 1. **API with just qmem to qurt\_mem name changed**

Table *2: qmem to qurt\_mem API function name changed only mapping*

|  |  |
| --- | --- |
| **Qmem API function Names** | **QuRT API function Names** |
| qmem\_region\_attr\_init | qurt\_mem\_region\_attr\_init |
| qmem\_region\_create | qurt\_mem\_region\_create |
| qmem\_region\_delete | qurt\_mem\_region\_delete |
| qmem\_region\_get\_attr | qurt\_mem\_region\_get\_attr |
| qmem\_kernel\_cache\_opt | qurt\_mem\_kernel\_cache\_opt |
| qmem\_kernel\_cache\_clean | qurt\_mem\_kernel\_cache\_clean |
| qmem\_cache\_flushall | qurt\_mem\_cache\_flushall |
| qmem\_cache\_clean | qurt\_mem\_cache\_clean |
| qmem\_cache\_clean\_aligning | qurt\_mem\_cache\_clean |
| qmem\_default\_pool | qurt\_mem\_default\_pool |
| qmem\_pool\_attach | qurt\_mem\_pool\_attach |

* 1. **API with just qtimer to qurt\_timer name changed**

Table *2: qmem to qurt\_mem API function name changed only mapping*

|  |  |
| --- | --- |
| **qtimer API function Names** | **QuRT API function Names** |
| qtimer\_cmd\_getattr | qurt\_timer\_cmd\_getattr |
| qtimer\_recover\_pc | qurt\_timer\_recover\_pc |
| qtimer\_cmd\_create | qurt\_timer\_cmd\_create |
| qtimer\_is\_init | qurt\_timer\_is\_init |
| qtimer\_cmd\_group | qurt\_timer\_cmd\_group |
| qtimer\_restart | qurt\_timer\_restart |
| qtimer\_callback | qurt\_timer\_callback |
| qtimer\_type | qurt\_timer\_type |
| qtimer\_attr\_init | qurt\_timer\_attr\_init |
| qtimer\_delete | qurt\_timer\_delete |
| qtimer\_group\_disable | qurt\_timer\_group\_disable |
| qtimer\_group\_enable | qurt\_timer\_group\_enable |
| qtimer\_sleep | qurt\_timer\_sleep |
| qtimer\_get\_attr | qurt\_timer\_get\_attr |
| qtimer\_cmd\_getticks | qurt\_timer\_cmd\_getticks |
| qtimer\_cmd\_free | qurt\_timer\_cmd\_free |
| qtimer\_stop | qurt\_timer\_stop |

* 1. **API function with name change other than blast to qurt**

Table *3: qmem to qurt\_mem API function name changed only mapping*

|  |  |
| --- | --- |
| **Interrupt functions** | |
| blast\_register\_fastint | qurt\_fastint\_register |
| blast\_deregister\_fastint | qurt\_fastint\_deregister |
| blast\_interrupt\_getconfig | qurt\_interrupt\_get\_config |
| blast\_get\_registered\_interrupts | qurt\_interrupt\_get\_registered |
| blast\_register\_interrupt | qurt\_interrupt\_register |
| blast\_deregister\_interrupt | qurt\_interrupt\_deregister |
| blast\_interrupt\_setconfig | qurt\_interrupt\_set\_config |
| blast\_ack\_interrupt | qurt\_interrupt\_acknowledge |
| **Memory functions** | |
| blast\_remove\_mapping | qurt\_mapping\_remove |
| blast\_create\_mapping | qurt\_mapping\_create |
| blast\_mem\_region\_attr\_getvirtaddr | qurt\_mem\_region\_attr\_get\_virtaddr |
| blast\_mem\_region\_attr\_setvirtaddr | qurt\_mem\_region\_attr\_set\_virtaddr |
| blast\_mem\_region\_attr\_getphysaddr | qurt\_mem\_region\_attr\_get\_physaddr |
| blast\_mem\_region\_attr\_setphysaddr | qurt\_mem\_region\_attr\_set\_physaddr |
| blast\_mem\_region\_attr\_setcachemode | qurt\_mem\_region\_attr\_set\_cache\_mode |
| blast\_mem\_region\_attr\_getcachemode | qurt\_mem\_region\_attr\_get\_cache\_mode |
| blast\_mem\_region\_attr\_setmapping | qurt\_mem\_region\_attr\_set\_mapping |
| blast\_mem\_region\_attr\_getmapping | qurt\_mem\_region\_attr\_get\_mapping |
| blast\_mem\_region\_attr\_getsize | qurt\_mem\_region\_attr\_get\_size |
| blast\_mem\_region\_attr\_gettype | qurt\_mem\_region\_attr\_get\_type |
| blast\_mem\_region\_attr\_settype | qurt\_mem\_region\_attr\_set\_type |
| blast\_mem\_region\_get\_attr | qurt\_mem\_region\_attr\_get |
| **Cache functions** | |
| blast\_mem\_cleaninv\_dcache | qurt\_mem\_dcache\_cleaninv |
| blast\_mem\_clean\_dcache | qurt\_mem\_dcache\_clean |
| blast\_mem\_inv\_dcache | qurt\_mem\_dcache\_inv |
| blast\_mem\_inv\_icache | qurt\_mem\_icache\_inv |
| **Error handler functions** | |
| blast\_reg\_error\_handler | qurt\_exception\_wait |
| blast\_exit | qurt\_exception\_raise\_nonfatal |
| blast\_fatal\_exit | qurt\_exception\_raise\_fatal |
| **Power control funtions** | |
| blast\_power\_shutdown\_enter | qurt\_power\_shutdown\_enter\_wrapper |
| **Mutex functions** | |
| blast\_rmutex\_trylock\_block\_once | qurt\_rmutex\_try\_lock\_block\_once |
| blast\_rmutex\_trylock | qurt\_rmutex\_try\_lock |
| blast\_mutex\_trylock | qurt\_mutex\_try\_lock |
| **Pimutex functions** | |
| blast\_pimutex\_trylock | qurt\_pimutex\_try\_lock |
| **Sem functions** | |
| blast\_sem\_trydown | qurt\_sem\_try\_down |
| **Thread funtions** | |
| blast\_thread\_get\_tid | qurt\_thread\_get\_timetest\_tid |
| blast\_thread\_set\_tid | qurt\_thread\_set\_timetest\_id |
| blast\_thread\_myid | qurt\_thread\_get\_id |
| blast\_thread\_set\_hw\_bitmask | qurt\_thread\_set\_affinity |
| blast\_prio\_set | qurt\_thread\_set\_priority |
| blast\_prio\_get | qurt\_thread\_get\_priority |
| **Tlb funtions** | |
| blast\_tlb\_mapping\_delete | qurt\_tlb\_delete\_mapping |
| blast\_tlb\_mapping\_create | qurt\_tlb\_create\_mapping |
| blast\_tlb\_setentry | qurt\_tlb\_set\_entry |
| blast\_tlb\_getentry | qurt\_tlb\_get\_entry |
| blast\_tlb\_query | qurt\_tlb\_query |
| blast\_tls\_key\_delete | qurt\_tls\_delete\_key |
| blast\_tls\_setspecific | qurt\_tls\_set\_specific |
| blast\_tls\_getspecific | qurt\_tls\_get\_specific |
| blast\_tls\_key\_create | qurt\_tls\_create\_key |
| **System clock funtions** | |
| blast\_system\_sclk\_attr\_gethwticks | qurt\_sysclock\_get\_hw\_ticks |
| qsystem\_sclk\_attr\_getexpiry | qurt\_sysclock\_get\_expiry |
| blast\_system\_sclk\_register | qurt\_sysclock \_register |
| blast\_system\_sclk\_alarm | qurt\_sysclock \_alarm\_create |
| blast\_system\_sclk\_timer | qurt\_sysclock \_timer\_create |
| **Timer funtions** | |
| blast\_timer\_getremaining | qurt\_timer\_get\_remaining |
| blast\_timer\_group\_enable | qurt\_timer\_group\_enable |
| blast\_timer\_group\_disable | qurt\_timer\_group\_disable |
| blast\_timer\_attr\_setduration | qurt\_timer\_attr\_set\_duration |
| blast\_timer\_attr\_getduration | qurt\_timer\_attr\_get\_duration |
| blast\_timer\_attr\_settype | qurt\_timer\_attr\_set\_type |
| blast\_timer\_attr\_gettype | qurt\_timer\_attr\_get\_type |
| blast\_timer\_attr\_setgroup | qurt\_timer\_attr\_set\_group |
| blast\_timer\_attr\_getgroup | qurt\_timer\_attr\_get\_group |
| blast\_timer\_attr\_setexpiry | qurt\_timer\_attr\_set\_expiry |

* 1. **API function with name change other than qmem to qurt\_mem**

*Table 4:Qmem to QuRT API function name mappings*

|  |  |
| --- | --- |
| **Qmem API function Names** | **QuRT API function Names** |
| qmem\_qurt\_cleaninv\_dcache | qurt\_mem\_cleaninv\_dcache |
| qmem\_qurt\_clean\_dcache | qurt\_mem\_clean\_dcache |
| qmem\_qurt\_inv\_dcache | qurt\_mem\_inv\_dcache |
| qmem\_qurt\_inv\_icache | qurt\_mem\_inv\_icache |

* 1. **API function with name change other than qtimer to qurt\_timer**

Table *2: qmem to qurt\_mem API function name changed only mapping*

|  |  |
| --- | --- |
| **qtimer API function Names** | **QuRT API function Names** |
| qtimer\_attr\_setgroup | qurt\_timer\_attr\_set\_group |
| qtimer\_attr\_getremaining | qurt\_timer\_attr\_get\_remaining |
| qtimer\_attr\_setexpiry | qurt\_timer\_attr\_set\_expiry |
| qtimer\_attr\_getgroup | qurt\_timer\_attr\_get\_group |
| qtimer\_attr\_settype | qurt\_timer\_attr\_set\_type |
| qtimer\_attr\_gettype | qurt\_timer\_attr\_get\_type |
| qtimer\_attr\_setduration | qurt\_timer\_attr\_set\_duration |
| qsystem\_sclk\_attr\_getticks | qurt\_timer\_get\_ticks |
| qsystem\_sclk\_attr\_getexpiry | qurt\_sysclock\_get\_expiry |
| qsystem\_sclk\_attr\_gethwticks | qurt\_sysclock\_get\_hw\_ticks |
| sclk\_timer\_create | qurt\_timer\_create |
| sclk\_timer\_stop | qurt\_timer\_stop |
| qsystem\_sclk\_attr\_getticks | qurt\_timer\_get\_ticks |
| qsystem\_sclk\_attr\_getexpiry | qurt\_sysclock\_get\_expiry |
| qsystem\_sclk\_attr\_gethwticks | qurt\_sysclock\_get\_hw\_ticks |

1. **Structure, typedef, constants Changes**

The constant and macros mapping from Blast to QuRT are sumarized in table 3

*Table 6: Blast to QuRT Constant and macros mappings*

|  |  |
| --- | --- |
| **Blast constant** | **QuRT constant** |
| BLAST\_MAX\_HTHREADS | QURT\_MAX\_HTHREADS |
| BLAST\_MEM\_CACHE\_FLUSH | QURT\_MEM\_CACHE\_FLUSH |
| BLAST\_MEM\_CACHE\_FLUSH\_INVALIDATE | QURT\_MEM\_CACHE\_FLUSH\_INVALIDATE |
| BLAST\_MEM\_CACHE\_NONE | QURT\_MEM\_CACHE\_NONE |
| BLAST\_MEM\_CACHE\_NONE\_SHARED | QURT\_MEM\_CACHE\_NONE\_SHARED |
| BLAST\_MEM\_CACHE\_WRITEBACK | QURT\_MEM\_CACHE\_WRITEBACK |
| BLAST\_MEM\_CACHE\_WRITEBACK\_NONL2CACHEABLE | QURT\_MEM\_CACHE\_WRITEBACK\_NONL2CACHEABLE |
| BLAST\_MEM\_CACHE\_WRITEBACK\_L2CACHEABLE | QURT\_MEM\_CACHE\_WRITEBACK\_L2CACHEABLE |
| BLAST\_MEM\_CACHE\_WRITETHROUGH | QURT\_MEM\_CACHE\_WRITETHROUGH |
| BLAST\_MEM\_CACHE\_WRITETHROUGH\_NONL2CACHEABLE | QURT\_MEM\_CACHE\_WRITETHROUGH\_NONL2CACHEABLE |
| BLAST\_MEM\_CACHE\_WRITETHROUGH\_L2CACHEABLE | QURT\_MEM\_CACHE\_WRITETHROUGH\_L2CACHEABLE |
| BLAST\_MEM\_ICACHE | QURT\_MEM\_ICACHE |
| BLAST\_MEM\_DCACHE | QURT\_MEM\_DCACHE |
| BLAST\_MEM\_CACHE\_INVALIDATE | QURT\_MEM\_CACHE\_INVALIDATE |
| BLAST\_MEM\_MAPPING\_NONE | QURT\_MEM\_MAPPING\_NONE |
| BLAST\_MEM\_MAPPING\_IDEMPOTENT | QURT\_MEM\_MAPPING\_IDEMPOTENT |
| BLAST\_MEM\_MAPPING\_PHYS\_CONTIGUOUS | QURT\_MEM\_MAPPING\_PHYS\_CONTIGUOUS |
| BLAST\_MEM\_MAPPING\_VIRTUAL | QURT\_MEM\_MAPPING\_VIRTUAL |
| BLAST\_MEM\_MAPPING\_VIRTUAL\_FIXED | QURT\_MEM\_MAPPING\_VIRTUAL\_FIXED |
| BLAST\_MEM\_MEMORY\_TCM | QURT\_MEM\_MEMORY\_TCM |
| BLAST\_MEM\_MEMORY\_DEFAULT | QURT\_MEM\_MEMORY\_DEFAULT |
| BLAST\_MEM\_REGION\_LOCAL | QURT\_MEM\_REGION\_LOCAL |
| BLAST\_MEM\_REGION\_SHARED | QURT\_MEM\_REGION\_SHARED |
| BLAST\_PERM\_EXECUTE | QURT\_PERM\_EXECUTE |
| BLAST\_PERM\_READ | QURT\_PERM\_READ |
| BLAST\_PERM\_FULL | QURT\_PERM\_FULL |
| BLAST\_PERM\_WRITE | QURT\_PERM\_WRITE |
| BLAST\_BARRIER\_OTHER | QURT\_BARRIER\_OTHER |
| BLAST\_BARRIER\_SERIAL\_THREAD | QURT\_BARRIER\_SERIAL\_THREAD |
| BLAST\_PMUCNT3 | QURT\_PMUCNT3 |
| BLAST\_PMUCNT2 | QURT\_PMUCNT2 |
| BLAST\_PMUCNT1 | QURT\_PMUCNT1 |
| BLAST\_PMUCNT0 | QURT\_PMUCNT0 |
| BLAST\_PMUCNT5 | QURT\_PMUCNT5 |
| BLAST\_PMUCNT4 | QURT\_PMUCNT4 |
| BLAST\_MAX\_TLS | QURT\_MAX\_TLS |
| BLAST\_MAX\_TLS\_INDEX | QURT\_MAX\_TLS\_INDEX |
| BLAST\_THREAD\_CFG\_BITMASK\_ALL | QURT\_THREAD\_CFG\_BITMASK\_ALL |
| BLAST\_EVAL | QURT\_EVAL |
| BLAST\_EOK | QURT\_EOK |

Table 5: Qmem to QuRT Constant and macros mappings

|  |  |
| --- | --- |
| **Qmem constant** | **QuRT constant** |
| QMEM\_MAPPING\_VIRTUAL | QURT\_MEM\_MAPPING\_VIRTUAL |
| QMEM\_MAPPING\_PHYS\_CONTIGUOUS | QURT\_MEM\_MAPPING\_PHYS\_CONTIGUOUS |
| QMEM\_MAPPING\_IDEMPOTENT | QURT\_MEM\_MAPPING\_IDEMPOTENT |
| QMEM\_MAPPING\_VIRTUAL\_FIXED | QURT\_MEM\_MAPPING\_VIRTUAL\_FIXED |
| QMEM\_MAPPING\_NONE | QURT\_MEM\_MAPPING\_NONE |
| QMEM\_CACHE\_NONE | QURT\_MEM\_CACHE\_NONE |
| QMEM\_CACHE\_FLUSH | QURT\_MEM\_CACHE\_FLUSH |
| QMEM\_CACHE\_INVALIDATE | QURT\_MEM\_CACHE\_INVALIDATE |
| QMEM\_CACHE\_FLUSH\_INVALIDATE | QURT\_MEM\_CACHE\_FLUSH\_INVALIDATE |
| QMEM\_CACHE\_WRITEBACK | QURT\_MEM\_CACHE\_WRITEBACK |
| QMEM\_CACHE\_NONE\_SHARED | QURT\_MEM\_CACHE\_NONE\_SHARED |
| QMEM\_CACHE\_WRITETHROUGH | QURT\_MEM\_CACHE\_WRITETHROUGH |
| QMEM\_CACHE\_WRITEBACK\_NONL2CACHEABLE | QURT\_MEM\_CACHE\_WRITEBACK\_NONL2CACHEABLE |
| QMEM\_CACHE\_WRITETHROUGH\_NONL2CACHEABLE | QURT\_MEM\_CACHE\_WRITETHROUGH\_NONL2CACHEABLE |
| QMEM\_CACHE\_WRITEBACK\_L2CACHEABLE | QURT\_MEM\_CACHE\_WRITEBACK\_L2CACHEABLE |
| QMEM\_CACHE\_WRITETHROUGH\_L2CACHEABLE | QURT\_MEM\_CACHE\_WRITETHROUGH\_L2CACHEABLE |
| QMEM\_ICACHE | QURT\_MEM\_ICACHE |
| QMEM\_DCACHE | QURT\_MEM\_DCACHE |
| QMEM\_PERM\_READ | QURT\_PERM\_READ |
| QMEM\_PERM\_WRITE | QURT\_PERM\_WRITE |
| QMEM\_PERM\_EXECUTE | QURT\_PERM\_EXECUTE |
| QMEM\_REGION\_LOCAL | QURT\_MEM\_REGION\_LOCAL |
| QMEM\_REGION\_SHARED | QURT\_MEM\_REGION\_SHARED |

Table 6: Qmem to QuRT typedef mappings

|  |  |
| --- | --- |
| **Qmem Typedefs** | **QuRT typedefs** |
| qmem\_pool\_t | qurt\_mem\_pool\_t |
| qmem\_perm\_t | qurt\_perm\_t |
| qmem\_cache\_mode\_t | qurt\_mem\_cache\_mode\_t |
| qmem\_cache\_type\_t | qurt\_mem\_cache\_type\_t |
| qmem\_cache\_op\_t | qurt\_mem\_cache\_op\_t |
| qmem\_region\_t | qurt\_mem\_region\_t |
| qmem\_region\_attr\_t | qurt\_mem\_region\_attr\_t |
| qmem\_region\_type\_t | qurt\_mem\_region\_type\_t |
| qmem\_pool\_t | qurt\_mem\_pool\_t |
| qmem\_perm\_t | qurt\_perm\_t |
| qmem\_mapping\_t | qurt\_mem\_mapping\_t |

|  |  |
| --- | --- |
| **Qtimer Typedefs** | **QuRT typedefs** |
| qtimer\_cmd\_free\_t | qurt\_timer\_cmd\_free\_t |
| qtimer\_attr\_t | qurt\_timer\_attr\_t |
| qtimer\_cmd\_t | qurt\_timer\_cmd\_t |
| qtimer\_cmd\_group\_t | qurt\_timer\_cmd\_group\_t |
| qtimer\_t | qurt\_timer\_t |
| qtimer\_type\_t | qurt\_timer\_type\_t |
| qtimer\_cmd\_create\_t | qurt\_timer\_cmd\_create\_t |
| qtimer\_duration\_t | qurt\_timer\_duration\_t |
| qtimer\_cmd\_getticks\_t | qurt\_timer\_cmd\_getticks\_t |
| qtimer\_time\_t | qurt\_timer\_time\_t |
| qtimer\_cmd\_getattr\_t | qurt\_timer\_cmd\_getattr\_t |
| qtimer\_callback\_t | qurt\_timer\_callback\_t |

Table 7 shows the typedef mapping from Blast to QuRT.

Table 7: Blast to QuRT Constant and macros mappings

|  |  |
| --- | --- |
| **Blast Typedefs** | **QuRT typedefs** |
| blast\_addr\_t | qurt\_addr\_t |
| blast\_app\_heap\_obj\_t | qurt\_app\_heap\_obj\_t |
| blast\_allsignal\_t | qurt\_allsignal\_t |
| blast\_anysignal\_t | qurt\_anysignal\_t |
| blast\_arch\_version\_t | qurt\_arch\_version\_t |
| blast\_barrier\_t | qurt\_barrier\_t |
| blast\_cache\_partition\_t | qurt\_cache\_partition\_t |
| blast\_cond\_t | qurt\_cond\_t |
| blast\_mem\_pool\_t | qurt\_mem\_pool\_t |
| blast\_mem\_cache\_type\_t | qurt\_mem\_cache\_type\_t |
| blast\_mem\_cache\_mode\_t | qurt\_mem\_cache\_mode\_t |
| blast\_mem\_cache\_op\_t | qurt\_mem\_cache\_op\_t |
| blast\_mem\_mapping\_t | qurt\_mem\_mapping\_t |
| blast\_mem\_region\_t | qurt\_mem\_region\_t |
| blast\_mem\_region\_attr\_t | qurt\_mem\_region\_attr\_t |
| blast\_mem\_region\_type\_t | qurt\_mem\_region\_type\_t |
| blast\_mutex\_t | qurt\_mutex\_t |
| blast\_perm\_t | qurt\_perm\_t |
| blast\_pipe\_data\_t | qurt\_pipe\_data\_t |
| blast\_pipe\_t | qurt\_pipe\_t |
| blast\_sem\_t | qurt\_sem\_t |
| BLAST\_utcb\_t | QURT\_utcb\_t |

Table 8 shows the global variables mapping from Blast to QuRT.

Table 8: Blast to QuRT global variable mappings

|  |  |
| --- | --- |
| **Blast Global Variables** | **QuRT Global Variables** |
| BLAST\_ugp\_ptr | QURT\_ugp\_ptr |
| BLAST\_UTCB | QURT\_UTCB |

Mapping the Blast functions returned codes to QuRT function sreturned codes are shown in table 9

Table 9 : sumarizes the API function returned codes

|  |  |
| --- | --- |
| **Blast functions returned codes** | **QuRT functions returned codes** |
| EOK | QURT\_EOK |
| EVAL | QURT\_EVAL |
| EMEM | QURT\_EMEM |
| EINVALID | QURT\_EINVALID |
| ENO\_MSGS | QURT\_ENOMSGS |
| EMAX\_MSGS | QURT\_EMAXMSGS |
| ENO\_MSGQ | QURT\_ENOMSGQ |
| EDUP\_MSGQ | QURT\_EDUPMSGQ |
| EFAILED | QURT\_EFAILED |
| ENOTALLOWED | QURT\_ENOTALLOWED |
| EDUP\_CLSID | QURT\_EDUPCLSID |
| EBADPARM | QURT\_EINVALID |
| EINVALIDITEM | QURT\_EINVALID |
| EBADHANDLE | QURT\_EINVALID |
| ENO\_INTERRUPTS | QURT\_ENOREGISTERED |
| EPC\_ISDB | QURT\_EISDB |
| EPC\_NOSTM | QURT\_ESTM |
| E\_TLS\_NOAVAIL | QURT\_ETLSAVAIL |
| E\_TLS\_NOENT | QURT\_ETLSENTRY |
| EINT | QURT\_EINT |
| ESIG | QURT\_ESIG |
| E\_OUTOFHEAP | QURT\_EHEAP |
| E\_MM\_OUT\_OF\_RANGE | QURT\_EMEMMAP |
| ENO\_THREAD | QURT\_ENOTHREAD |
| E\_L2CACHABLE\_NOT\_SUPPORTED | QURT\_EL2CACHE |
| E\_FATAL | QURT\_EFATAL |
| E\_INT\_DEREGISTER | QURT\_EDEREGISTERED |
| E\_TLB\_CREATE\_SIZE | QURT\_ETLBCREATESIZE |
| E\_TLB\_CREATE\_UNALIGNED | QURT\_ETLBCREATEUNALIGNED |

Table10: Blast to QuRt demand paging events mapping

|  |  |
| --- | --- |
| **Blast demand paging events** | **QuRt demand paging events** |
| EVENT\_PAGEFAULT | QURT\_EVENT\_PAGEFAULT |
| EVENT\_SYSTEM\_ERR | QURT\_EVENT\_SYSTEM\_ERR |
| EVENT\_SUSPEND | QURT\_EVENT\_SUSPEND |

Table10: Blast to QuRt Env object events mapping

|  |  |
| --- | --- |
| **Blast Env object events** | **QuRt Env object events** |
| ENV\_OBJ\_SWAP\_POOLS | QURT\_ENV\_OBJ\_SWAP\_POOLS |
| ENV\_OBJ\_APP\_HEAP | QURT\_ENV\_OBJ\_APP\_HEAP |
| ENV\_OBJ\_TIMER | QURT\_ENV\_OBJ\_TIMER |
| ENV\_OBJ\_ARCH\_VER | QURT\_ENV\_OBJ\_ARCH\_VER |

1. **Libs file changes**

Some lib names are changes as

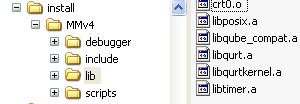


Table 10: Blast to QuRT Libs name mapping

|  |  |
| --- | --- |
| **Blast libs** | **QuRT libs** |
| libposix.a | libposix.a |
| libqube\_compat.a | libqube\_compat.a |
| libblast.a | libqurt.a |
| libblastkernel.a | libqurtkernel.a |
| libtimer.a | libtimer.a |

**Appendix A.** Detail of structure, typedef changed

**UTCB change**

|  |  |
| --- | --- |
| **Blast UTCB** | **QuRT UTCB** |
| unsigned int thread\_id | unsigned int thread\_id |
| void (\*entrypoint)(void \*) | void (\*entrypoint)(void \*) |
| void \*arg | void \*arg |
| long long int thread\_name0 |  |
| long long int thread\_name1 |  |
|  | qurt\_thread\_attr\_t attr |
| unsigned int stack\_size |  |
| qurt\_anysignal\_t anysignal | qurt\_anysignal\_t anysignal |

**New qurt\_thread\_attr structure**

typedef struct \_qurt\_thread\_attr {

char name[QURT\_THREAD\_ATTR\_NAME\_MAXLEN]; /\*\*< Thread name \*/

unsigned char tcb\_partition; /\*\*< Should the thread TCB reside in RAM or

on chip memory (i.e. TCM) \*/

unsigned char affinity; /\*\*< HW bitmask indicating the threads it can run on \*/

unsigned short priority; /\*\*< Thread priority \*/

unsigned short asid; /\*\*< Address space ID \*/

unsigned short timetest\_id; /\*\*< TIMETEST ID \*/

unsigned int stack\_size; /\*\*< Thread's stack size \*/

void \*stack\_addr; /\*\*< Stack address base.

(stack\_addr, stack\_addr+stack\_size-1)

is the range of the stack \*/

} **qurt\_thread\_attr\_t**;

**Appendix B.**

Lpass CRM build Porting example

Lpass Blast version in CRM build is 2.4.2

Qurt released version

Referenced build

[\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110](file:///\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437_qurt110)

To integrate QuRT to 8960 Lpass CRm build, following these changes

1. Rename directory blast to qurt

[\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\lpass\_proc\blast](file:///\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437_qurt110\lpass_proc\blast)

to

[\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\lpass\_proc\qurt](file:///\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437_qurt110\lpass_proc\qurt)

1. Change the lpass\_proc\makefile

[\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\lpass\_proc\makefile](file:///\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437_qurt110\lpass_proc\makefile)

Replace

* blast => qurt
* BLASTK => QURTK
* BLAST => QURT
* blastk => qurtk
* change to blast\_config.py qurt\_config.py

example: chmod -R 777 qurt/config/8960

python $(Q6\_RTOS\_ROOT)/install/MMv4/scripts/qurt\_config.py update -T $(Q6\_TOOLS\_ROOT) -o $(CUST\_CONFIG\_DIR)

1. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\build\Make\_Defs

replaces “blast” by “qurt”

CUST\_CONFIG\_DIR=qurt/config/8960

1. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\build\Make\_Ar
2. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\elite.linker
3. L:\M8960AAAAANAAL100205\lpass\_proc\dal\framework\Makefile
4. …\drivers\InterruptController\src\blast\daldrvintrctrl.min

vpath %.h $(PLATFORM\_ROOT)/blast/install/include

1. …\framework\sys\src\blast\dalsys.min

vpath % $(PLATFORM\_ROOT)/blast/install/include/qube

vpath % $(PLATFORM\_ROOT)/blast/install/include/posix

vpath % $(PLATFORM\_ROOT)/blast/install/include

1. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\DalOverrides\defines\_blast.min

Add qube path

RTOSINC = $(ROOT)/qurt/install/$(TARGET)/include#

add >>

RTOSQUBEINC = $(ROOT)/qurt/install/$(TARGET)/include/qube#

CC\_INC= \

-I$(QDSP6GCCINC) \

-I$(QDSP6QUBEINC) \

add >>

-I$(RTOSQUBEINC) \

1. Api function parameters changed

corebsp\power\sleep\src\asic\8660\lpa\lpr\_definition\_cpu\_vdd.c

>>> void)blast\_power\_shutdown(); =>blast\_power\_shutdown(0);

1. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\corebsp\services\osal\blast\elite.mak

Blast => qurt

INCLUDES :=-I. \

-I$(ROOT)/corebsp/services/corebsp\_utils/src \

-I$(ROOT)/corebsp/services/common/inc \

-I$(ROOT)/corebsp/services/osal/inc \

-I$(ROOT)/qurt/install/$(TARGET)/include \

-I$(ROOT)/qurt/install/$(TARGET)/include/qube \

1. Add #include “blast.h”

.../../../corebsp/services/corebsp\_utils/src/queue.h

>>> ../../../corebsp/services/dsm/src/dsm\_lock.c

Replaces #include "blast\_error.h" => #include "blast.h"

../../../corebsp/services/dsm/src/dsm\_lock.h

#include "blast.h"

#include "qube.h"

1. change blast => qurt

\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\corebsp\services\dsm\elite.mak

INCLUDES :=-I. \

-I./inc \

-I./src \

-I$(ROOT)/corebsp/services/corebsp\_utils/src \

-I$(ROOT)/corebsp/services/common/inc \

-I$(ROOT)/corebsp/services/diag/inc \

> >> -I$(ROOT)/qurt/install/$(TARGET)/include \

>>> -I$(ROOT)/qurt/install/$(TARGET)/include/qube \

1. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\corebsp\services\diag\elite.mak

-I$(ROOT)/corebsp/services/err/src \

>> -I$(ROOT)/ qurt /install/$(TARGET)/include \

>> -I$(ROOT)/ qurt install/$(TARGET)/include/qube \

>> -I$(ROOT)/ qurt /install/$(TARGET)/include/posix \

-I./inc \

-I./src

1. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\corebsp\power\sleep\elite.mak

>> -I$(ROOT)/qurt/install/$(TARGET)/include/qube \

>> -I$(ROOT)/qurt/install/$(TARGET)/include/posix \

1. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\corebsp\services\dsm\Elite.mak:

>>> -I$(ROOT)/qurt/install/$(TARGET)/include \

>>> -I$(ROOT)/qurt/install/$(TARGET)/include/qube \

1. L:\M8960AAAAANAAL100205\lpass\_proc\elite\common\Hexos\Elite.mak:

L:\M8960AAAAANAAL100437\_qurt110\lpass\_proc\elite\common\qurt\_elite\Elite.mak

>>> -I$(ROOT)/qurt/config/8660 \

L:\M8960AAAAANAAL100437\_qurt110\lpass\_proc\elite\common\qurt\_elite

1. Add include “blast.h”

- ./src/osal\_blast.c:

>>> include blast.h

- L:\ M8960AAAAANAAL100437\_qurt110\lpass\_proc\corebsp\services\osal\blast\inc\ osal\_blast\_types.h

>> #include "qube.h"

>> #include "blast.h"

1. L:\ M8960AAAAANAAL100437\_qurt110\lpass\_proc\Audio\midi\QSynth\QSynthCoreLib\src\synth.cpp

./src/synth.cpp:42:27: error: blast\_fastint.h: No such file or directory

replace by #include "blast.h" (which include qurt.h)

1. Add include “qube.h”

. ../../../corebsp/services/corebsp\_utils/src/queue.h

#include "blast.h"

Add >> #include "qube.h"

../../../corebsp/services/dsm/src/dsm\_lock.h: #include "blast.h"

Add >> #include "qube.h"

1. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\elite\common\ qurt\_elite\Elite.mak

L:\M8960AAAAANAAL100205\lpass\_proc\elite\common\Hexos\Elite.mak

change blast => qurt

INCLUDES :=-I. \

-I./inc \

>> -I$(ROOT)/qurt/config/8660 \

-I$(ROOT)/Audio/drivers/hw/avtimer/hal/inc \

-I$(ROOT)/Audio/drivers/hw/avtimer/avtimer\_driver/inc \

L:\M8960AAAAANAAL100205\lpass\_proc\elite\common\Hexos\inc\Hexos.h

\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\elite\common\qurt\_elite\inc\qurt\_elite.h

\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\EliteCore.mak

\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\elite.mak

INCLUDES :=\

...

-I$(ROOT)/blast/install/$(TARGET)/include/qube \

-I$(ROOT)/blast/install/$(TARGET)/include/posix \

Change

>> -I$(ROOT)/qurt/install/$(TARGET)/include/qube \

>> -I$(ROOT)/qurt/install/$(TARGET)/include/posix \

ifeq ($(INCLUDE\_STRIP\_LIBS), YES)

PRELINK +=$(RTOS\_LIBS\_ROOT)/strip/crt0.o \

>> $(RTOS\_LIBS\_ROOT)/strip/libqurt.a \

>> $(RTOS\_LIBS\_ROOT)/strip/libqurtkernel.a \

$(RTOS\_LIBS\_ROOT)/strip/libqube\_compat.a \

$(RTOS\_LIBS\_ROOT)/strip/libtimer.a \

$(CUST\_CONFIG\_DIR)/cust\_config.o

else

PRELINK +=$(RTOS\_LIBS\_ROOT)/crt0.o \

>> $(RTOS\_LIBS\_ROOT)/libqurt.a \

"" $(RTOS\_LIBS\_ROOT)/libqurtkernel.a \

$(RTOS\_LIBS\_ROOT)/libqube\_compat.a \

$(RTOS\_LIBS\_ROOT)/libtimer.a \

$(CUST\_CONFIG\_DIR)/cust\_config.o

endif

1. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\Audio\midi\QSynth\QSynthCoreLib\src\synth.cpp

#include "qube.h"

Replaced >> #include "blast\_fastint.h" => #include "blast.h"

1. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\build/make\_defs

Replaced

export LIB\_DALLIB\_ROOT := $(ROOT)/deploypkg/libkit/lib/blast\_$(V\_FLAVOR)

by

>> export LIB\_DALLIB\_ROOT := $(ROOT)/deploypkg/libkit/lib/qurt\_$(V\_FLAVOR)

1. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\corebsp\services\diag\platform\adsp\src\diagstub.c

>>replace "qube.h" by #include "blast.h"

1. L:\M8960AAAAANAAL100205/lpass\_proc/Audio/drivers/hw/dml/dml/src/DMLite.c

\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \_qurt\lpass\_proc\Audio\drivers\hw\dml\common\src\ DMLite.c

#if defined(DML\_USE\_QUBE)

add >> #include "blast.h"

#include "qube.h"

#endif

1. Need to take the crm build

\\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\qurt\kernel\scripts\Input\default\_build\_config.def

1. Rename/copy \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\ q6\_blast\_config.xml to q6\_qurt\_config.xml
2. \\qcdfs\qct\qctdata\LinuxEngBuilds\users11\opham\M8960AAAAANAAL100437\_qurt110\l \lpass\_proc\dal\drivers\InterruptController\src\blast\ DALInterruptController.c

blast\_thread\_set\_name(name0, name1);

ret = qurt\_thread\_create(&config->thread\_id, &attr, ist\_main, (void \*)NULL;

1. L:\M8960AAAAANAAL100437\_qurt110\lpass\_proc\elite\common\qurt\_elite\inc\qurt\_elite.h

#if defined(\_\_qdsp6\_\_)

#include "blast.h"

#include "qube.h"

#include "q6protos.h"

#include "sched\_config.h"

#include "q6\_system\_defns.h"

#endif

1. L:\M8960AAAAANAAL100437\_qurt110\lpass\_proc\elite\common\qurt\_elite\src\ qurt\_elite\_thread.cpp

Blast\_thread\_creat() problem

Add

typedef void (\*qurt\_elite\_thread\_handler\_func\_t) (void \*);

change

static void \* qurt\_elite\_thread\_stub(qurt\_elite\_thread\_list\_t \*pThreadNode);

return NULL

to

static void qurt\_elite\_thread\_stub(qurt\_elite\_thread\_list\_t \*pThreadNode)

return

1. L:\M8960AAAAANAAL100437\_qurt110\lpass\_proc\qdsp6\wdt\src\ wdt.c

Remove #include "blast\_mutex.h"

ret = qurt\_thread\_create(…)

1. L:\M8960AAAAANAAL100437\_qurt110\lpass\_proc\corebsp\power\sleep\src\asic\8960\lpa\lpr\_definition\_cpu\_vdd.c

Replace #include "blast\_power.h"

By #include "blast.h”

1. L:\M8960AAAAANAAL100437\_qurt110\lpass\_proc\corebsp\power\sleep\src\os\blast\lpa\sleep\_os.cpp

#include "blast.h"

Delete>> #include "blast\_error.h"

Delete>> #include "blast\_power.h"

#include "CoreVerify.h"

#include "CoreList.h"

#include "npa.h"

Delete>> #include "qthread.h"

Delete>> #include "qtimer.h"

1. L:\M8960AAAAANAAL100437\_qurt110\lpass\_proc\corebsp\power\utils\inc\CoreTime.h

#ifdef USES\_BLAST

#include "blast.h"

Delete >>#include "atomic\_ops.h"

#include "qube.h"

Delete >> #include "qtimer.h"

#elif (defined (WINSIM) || defined (WIN\_DEBUGGER\_EXTENSION))

#include "windows.h"

#elif defined(TARGET\_UEFI)

#include "TimetickUEFI.h"

extern uint32 UEFITimeGet(void);

#elif !defined(FEATURE\_POWER\_UTIL\_DAL\_TIMETICK)

#include "timetick.h"

#endif

1. L:\M8960AAAAANAAL100437\_qurt110\lpass\_proc\corebsp\power\sleep\src\os\blast\lpa\ sleep\_os.cpp

char cname[16]="SLEEP";

qurt\_thread\_attr\_set\_name(&attr,cname );

qurt\_thread\_attr\_t attr;

qurt\_thread\_t tid;

typedef void (\*sleep\_handler\_func\_t) (void \*);

qurt\_thread\_attr\_init (&attr);

qurt\_thread\_attr\_set\_stack\_size (&attr, SLEEP\_STACK\_SIZE);

qurt\_thread\_attr\_set\_stack\_addr (&attr, sleep\_stack);

qurt\_thread\_attr\_set\_priority (&attr, (unsigned short)0xFF);

ret = qurt\_thread\_create(&tid, &attr, (sleep\_handler\_func\_t)sleep\_task, NULL);

1. L:\M8960AAAAANAAL100437\_qurt110\lpass\_proc\corebsp\services\diag\DCM\common\src\diag.c

ret = qurt\_thread\_create(…);

1. M8960AAAAANAAL100437\_qurt110/lpass\_proc/apr/target/adsp/config/../../../target/adsp/utils/blast2/src/apr\_thread.c:
2. M8960AAAAANAAL100437\_qurt110/lpass\_proc/corebsp/services/osal/blast/src/osal\_blast.c

qurt\_thread\_attr\_set\_name(&attr,cname );

typedef void (\* thread\_handler\_func\_t) (void \*);

ret = qurt\_thread\_create(&tid, &attr, (\* thread\_handler\_func\_t)<thread\_hander>, NULL);

1. /prj/qct/asw/engbuilds/sd/users11/opham/M8960AAAAANAAL100437\_qurt110/lpass\_proc/corebsp/services/diag/DCM/common/src/diag.c:2194: undefined reference to `osal\_set\_task\_name'