```cpp
#include <tapasco.hpp>
/* Main Abstraction of a Tapasco Device
    Tapasco tapasco; // Acquire Device 0
*/
class Tapasco {};
// Launch a job
job_future launch(tapasco_kernel_id_t const k_id, RetVal<R>& ret,
↪ Targs... args)
job_future launch(tapasco_kernel_id_t const k_id, Targs... args)
// Call future to wait for job completion

// Type specifiers for blocks of data
OutOnly<T> makeOutOnly(T &t)
InOnly<T> makeInOnly(T &t)
WrappedPointer<T> makeWrappedPointer(T *t, size_t sz)
```

- Simple HLS Kernel

```
b[i] = a[i] * 5;
```

- TaPaSCo API

```
Tapasco tapasco;
auto a_wrapped = makeWrappedPointer(a.data(), a.size());
auto b_wrapped = makeWrappedPointer(b.data(), b.size());
auto job = tapasco.launch(SIMPLE_HLS_ID,
↪  makeInOnly(a_wrapped), makeOutOnly(b_wrapped));
job(); // Wait for completion
processResult(b);
```