How to use arrays of interfaces

version	1.0.0
scope	Example. This code is provided as example code for a user to base their code on.
description	How to use arrays of interfaces
boards	Unless otherwise specified, this example runs on the SliceKIT Core Board, but can easily be run on any XMOS device by using a different XN file.

Arrays of interfaces can be declared to provide a multiple connections between tasks. Individual elements of the array can be passed to tasks or the whole array can be passed on. In the following example, the array is passed to `task3` which connects to both `task1` and `task2`.

```
int main() {
  interface if1 c[2];
  par {
    task1(c[0]);
    task2(c[1]);
    task3(c, 2);
  }
  return 0;
}
```

The tasks given an element of the array use the interface end as normal.

```
void task1(client interface if1 c)
{
  c.msg(5);
}
```

The server end can select over the entire array by using a pattern variable in the case within the select. The syntax is:

```
case c[int i].msg(int x):
   // handle the case
   ...
   break;
```

Here, the variable `i` is declared as a subscript to the array `c` which means that the case will select over the entire array and wait for a message from one of the elements.

When a message is received, `i` is set to the index of the array element that the message came in on.

```
void task3(server interface if1 c[n], unsigned n)
{
  // receive two messages
  for (int i = 0;i < 2; i++) {
    select {
    case c[int i].msg(int x):
      printf("Received value %d from connection %d\n", x, i);
      break;
    }
  }
}
```

XMOS