

Application Note: AN10083

How to use variable length arrays

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to use variable length arrays.

Required tools and libraries

This application note is based on the following components:

- xTIMEcomposer Tools - Version 14.0.0

Required hardware

Programming how-tos are generally not specific to any particular hardware and can usually run on all Xmos devices. See the contents of the note for full details.

1 How to use variable length arrays

In xC, array declarations need to be a constant size. The exception to this is a local array that can be declared as a variable size based on a parameter, provided that parameter is marked both `static` and `const`:

```
void f(static const int n)
{
    printf("Array length = %d\n", n);

    int arr[n];
    for (int i = 0; i < n; i++) {
        arr[i] = i;
        for (int j = 0; j < i; j++) {
            arr[i] += arr[j];
        }
    }

    printf("-----\n");
    for (int i = 0; i < n; i++) {
        printf("Element %d of arr is %d\n", i, arr[i]);
    }
    printf("-----\n\n");
}
```

When calling functions with static parameters, the argument has to be either:

- a constant expression
- a static const parameter to the caller function

```
void g(static const int n)
{
    // static parameter can be called with a constant expression argument
    f(2);
    // or passing on a static const parameter
    f(n);
}
```

These restrictions mean that the compiler can still statically track stack usage despite the local array having variable size.