Global Variables

When you create a variable, it can only be used inside the function or task where it was declared. This can be a problem when you need to use the same variable in several different places – for example, a function you made, **and** task main.

Scope

Variables exist only within certain boundaries, for example, only within the functions where they are declared. Scope is a definition of these boundaries: **how broadly applicable (or "visible") a value or variable is**.

Scope exists to prevent conflicts between functions with similarly-named variables, and (more importantly) to keep variables from different functions from accidentally interfering with each other, or even with themselves if the same function is run more than once.

In general, the rule is that a variable can only be used **within the task or function where it is declared**, including task main. If you try to use a variable in location outside its scope, ROBOTC will give you an error that no such variable exists, because the program at that point cannot "see" it.

Global Variables

One very rough way to get around the limitations of scope in a program is to declare a variable as **global**. A global variable is declared **outside** any functions or tasks, and therefore typically appears at the very top of a program.

Because they are declared at a level broader than any task or function, **all** functions and tasks can "see" global variables, and they do not lose their value even after a function or task ends. This is both a strength and a liability.

1. Declare global variable
The global variable is declared outside any tasks or functions, allowing them all to see it.

2. Variable is "visible" inside functions
The global variable can be used and changed inside functions.

3. Variable is "visible" inside tasks
The global variable can be used and changed inside tasks such as task main.

```
int timeValue;

void changeValue()
{
  timeValue = 6000;
}

task main()
{
  timeValue = 2000;
  changeValue();
  wait1Msec(timeValue);
}
```

A Hint of Trouble

How long will this program wait at the end, 2000 or 6000ms? Because both the function and the task main were able to modify the value of this variable, it is more difficult to tell what its value is by the time it is used. As your program gets more complex, excessive use of global variables may lead to more and more confusion.