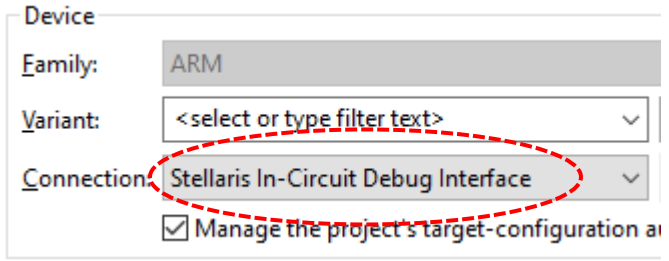
Debugging

SDU

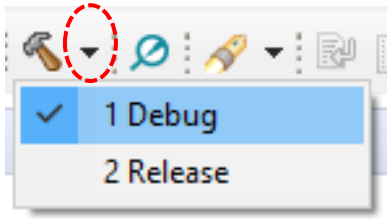
Debugging

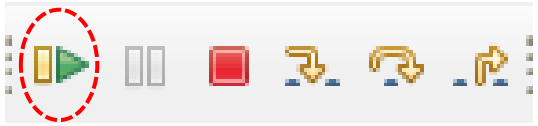
- Debugging while the code is running on the actual device
- Stellaris ICDI – in-circuit debug interface
- Make sure Stellaris ICDI is selected for your project
  - Project -> Properties -> CCS General

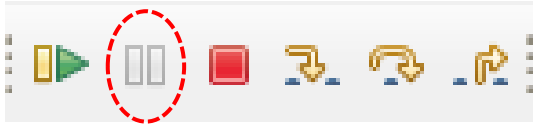


Debugging (continued)

- Build Debug version
- Start the Debugger (enter into debug mode)
- Start running the code in the debugger
- Pause (only possible when running)
- Stop

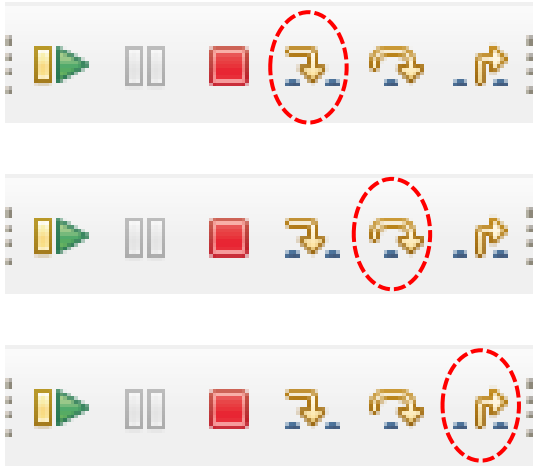






Debugging (continued)

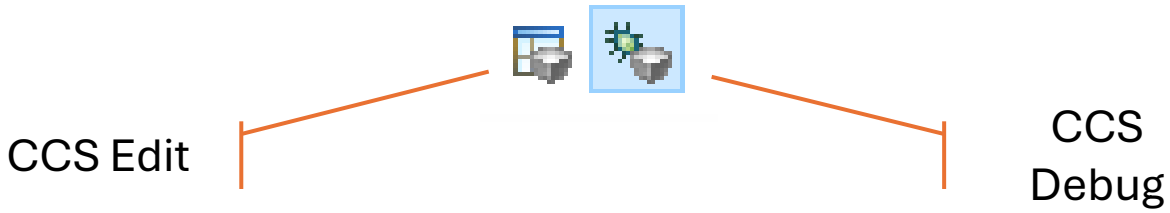
- Step by step debugging
  - Step into (function)
  - Step over line
  - Step out of (function)
- Stop



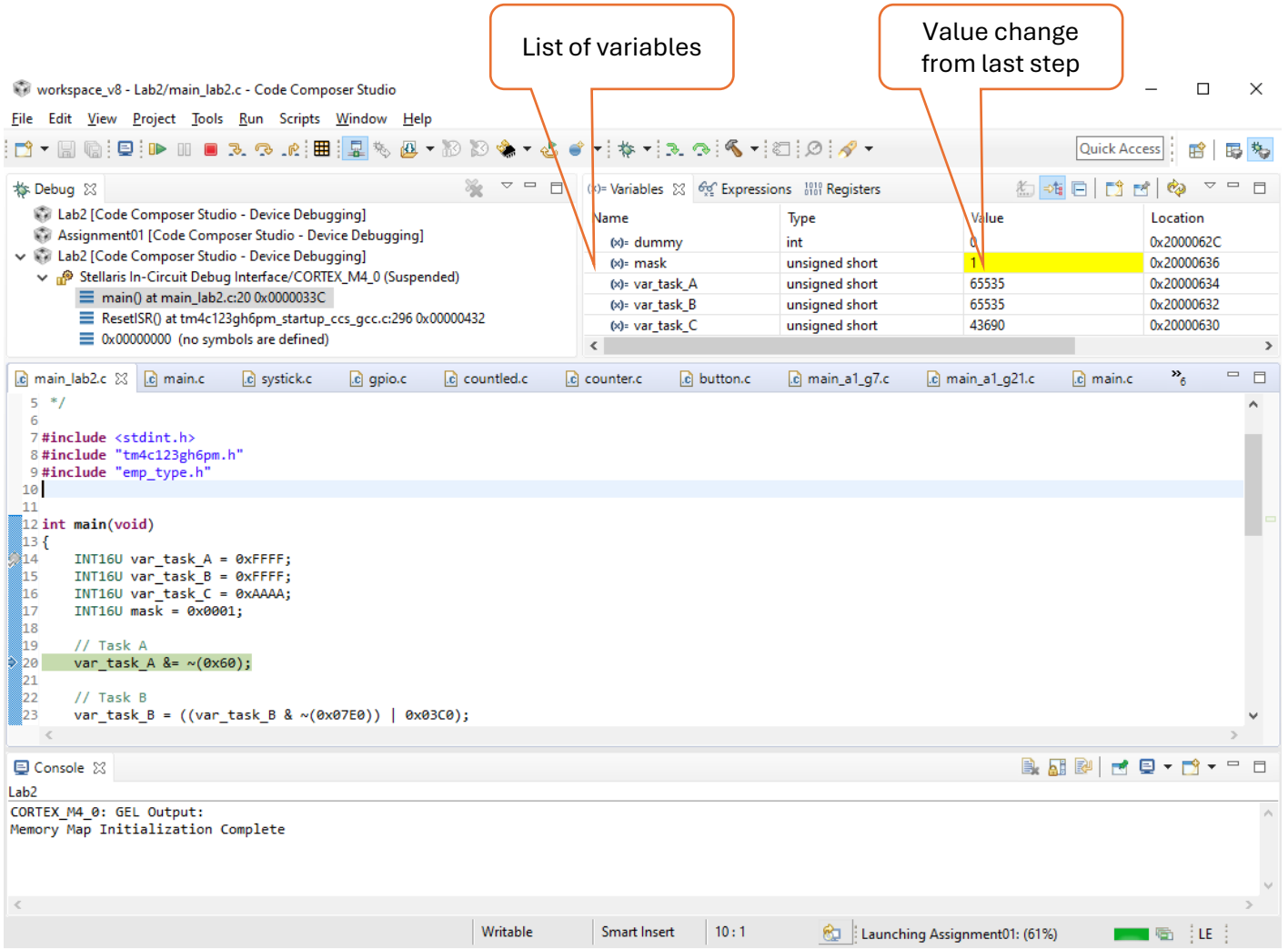
SDU

Debugging (continued)

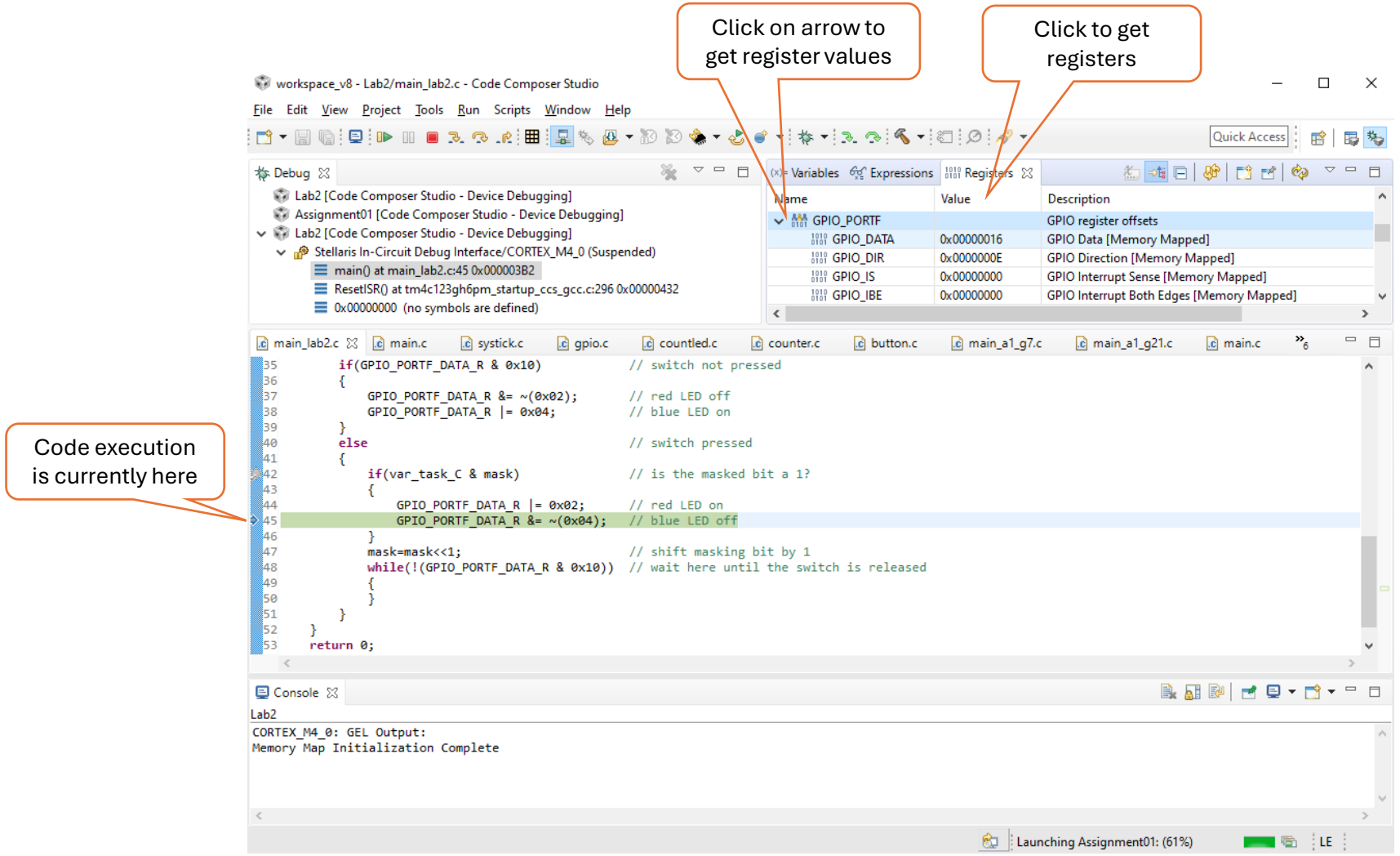
- Switch between Editor and Debug modes (upper right corner)



Debugging (continued)



Debugging (continued)



Debugging - breakpoints

Double click here to create a breakpoint

```
35          if(GPIO_PORTF_DATA_R & 0x10)            // switch not pressed
36          {
37              GPIO_PORTF_DATA_R &= ~(0x02);       // red LED off
38              GPIO_PORTF_DATA_R |= 0x04;          // blue LED on
39          }
40          else                                    // switch pressed
41          {
42              if(var_task_C & mask)               // is the masked bit a 1?
43              {
44                  GPIO_PORTF_DATA_R |= 0x02;      // red LED on
45                  GPIO_PORTF_DATA_R &= ~(0x04);   // blue LED off
46              }
47              mask=mask<<1;                       // shift masking bit by 1
48              while(!(GPIO_PORTF_DATA_R & 0x10))  // wait here until the switch is released
49              {
50              }
51          }
52      }
53      return 0;
```

Breakpoint created

```
35          if(GPIO_PORTF_DATA_R & 0x10)            // switch not pressed
36          {
37              GPIO_PORTF_DATA_R &= ~(0x02);       // red LED off
38              GPIO_PORTF_DATA_R |= 0x04;          // blue LED on
39          }
40          else                                    // switch pressed
41          {
42              if(var_task_C & mask)               // is the masked bit a 1?
43              {
44                  GPIO_PORTF_DATA_R |= 0x02;      // red LED on
45                  GPIO_PORTF_DATA_R &= ~(0x04);   // blue LED off
46              }
47              mask=mask<<1;                       // shift masking bit by 1
48              while(!(GPIO_PORTF_DATA_R & 0x10))  // wait here until the switch is released
49              {
50              }
51          }
52      }
53      return 0;
```

Debugging - breakpoints

- Now when you resume debug execution  , it will automatically stop once it hits the defined breakpoint
- Try it out

SDU