8.7 Advanced topics threading memory management templates

8.7.1 Bit-field variable type

Variable bit fields are a specific type of data structure in C++ that allows a user to store multiple bit-sized values within a single variable. This can be useful for storing several different values in the same memory space or for compressing data. An example of a variable bit field in C++ is shown below:

```
struct example {
 unsigned int value1 : 4; // Use 4 bits
 unsigned int value2 : 8; // Use 8 bits
 unsigned int value3 : 12; // Use 12 bits
} myStruct;
```

In this example, we have defined a structure called ‘example’ which contains three members - ‘value1’, ‘value2’, and ‘value3’. Each of these members has been defined as a variable bit field using the ‘unsigned int’ data type and the ‘:’ syntax, which allows us to specify the number of bits that each member should use. In this case, ‘value1’ will use 4 bits, ‘value2’ will use 8 bits, and ‘value3’ will use 12 bits. To access these values, we can use the members of the structure, for example, ‘myStruct.value2’.

```
struct adc4 {
  unsigned int value1 : 10;
  unsigned int value2 : 10;
  unsigned int value3 : 10;
  unsigned int value4 : 10;
};

unsigned int adc_val[40];           //40 values
adc4 myAdc[10];                     //40 values

void setup() {
  Serial.begin(9600);
  Serial.println(sizeof(adc_val));  //print 80
  Serial.println(sizeof(myAdc));    //print 50
}

void loop() {

}
```