Memory sections help manage program memory (Flash) and data memory (RAM). Here are the common memory sections in microcontrollers:

|  |  |  |
| --- | --- | --- |
| 1 | **Text Section (Code Section):** The "Text" section, also known as the "Code" section, stores the machine code or program instructions that the microcontroller executes. It typically resides in the Flash memory. | Example:  int main() { // Code in the "Text" section } |
| 2 | **Read-Only Data Section (rodata)** The "Read-Only Data" section, often labelled as "rodata," stores data that should not be modified during program execution. This can include constants, lookup tables, and other read-only data. | Example: const int constantValue = 42; // Stored in "rodata" |
| 3 | **Data Section (RAM Data)** The "Data" section stores variables and data that need to be read from and written to during program execution. It is typically located in the microcontroller's RAM. | Example: int counter = 0; // Stored in the "Data" section (RAM) |
| 4 | **BSS Section (Uninitialized Data)**  The "Block Started by Symbol" (BSS) section contains uninitialized global and static variables. It is set to zero by default. This section is often part of the RAM. | Example: int uninitializedVar; // Stored in the "BSS" section (RAM) |
| 5 | **Common Section**  The "Common" section is used for global variables when multiple source files reference the same variable name but define it in different places. The linker resolves these references and allocates a single memory location for the variable. | Example: - In file1.c: int sharedVar = 42; // Define the variable  - In file2.c: extern int sharedVar; // Declare the variable from file1.c |

