how much allows CMSIS and HAL layers on MCU (example stm32) to write code portable between MCUs of different manufacturers?

The CMSIS (Cortex Microcontroller Software Interface Standard) and HAL (Hardware Abstraction Layer) libraries are designed to help developers write portable and reusable code across different microcontroller families, primarily for ARM Cortex-based microcontrollers.

CMSIS is a vendor-independent hardware abstraction layer for the Cortex-M processor series, which provides a common programming interface for ARM Cortex-M based microcontrollers from different manufacturers. This helps in writing portable and reusable code, as the same code can be used across different Cortex-M based MCUs with minimal modifications.

However, CMSIS alone does not guarantee complete portability across different MCU manufacturers, as each manufacturer usually provides their own specific set of peripheral drivers and middleware components. That's where HAL comes in.

HAL is a higher-level, vendor-specific library that provides an additional abstraction layer for peripheral drivers and middleware components. For example, STM32 microcontrollers have their own HAL libraries called STM32Cube, which is specific to STMicroelectronics' STM32 series of microcontrollers.

While HAL libraries help in writing portable code within the same family of microcontrollers (e.g., different STM32 series), they are not designed to be completely portable across different manufacturers. To achieve that, you may need to use third-party libraries or write your own abstraction layer that can work with different HALs from different manufacturers.

In summary, CMSIS and HAL layers allow for a certain level of code portability across different Cortex-M based microcontrollers, but you may still need to make adjustments when switching between manufacturers. To achieve complete portability, you may need to write your own abstraction layer or use third-party libraries that can interface with various HALs.