

# 4. Assembly Integration with Arduino IDE

To combine Assembly with Arduino C++ code, the `extern "C"` directive is used in the `.ino` file and the `.global` directive is used in the `.S` (Assembly) file.

## File Structure:

### .ino File (C/C++):

```
extern "C" {
    void start();    // Declaration of function defined in Assembly
    void loop_asm(); // Another function from Assembly
}

void setup() {
    start();        // Call Assembly function for initialization
}

void loop() {
    loop_asm();    // Call Assembly function for main loop
}
```

### .S File (Assembly):

```
#define __SFR_OFFSET 0x00
#include "avr/io.h"

.global start
.global loop_asm

start:
    SBI DDRB, 5    ; Set PB5 (Pin 13) as Output
    RET           ; Return to caller

loop_asm:
```

```
SBI PORTB, 5 ; Turn on LED
; ... other code
RET
```

## Directive Explanations:

- `#define __SFR_OFFSET 0x00`: Sets the offset for I/O registers to use symbolic names (DDRB, PORTB, etc.).
- `#include "avr/io.h"`: Includes register definitions for the AVR chip.
- `.global`: Makes label/function accessible from other files (exported symbol).
- `RET`: Instruction to return from subroutine to the calling program.

---

Revision #1

Created 2026-02-04 06:51:55 UTC by AX

Updated 2026-02-04 06:53:14 UTC by AX