

Flash - FlashDemo

Example Code

```
#include <Flash.h>

#define SWAP_SERIAL

#define BASE_ADDRESS (PROGMEM_SIZE - (0x800))
// This is 2048 bytes before the end of the flash

uint16_t testArray[] = {0xBADD, 0xBEEF, 0xD00D};
char testArray2[15] = "Hello Flash";
uint8_t testArray3[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  // Blink the LED for good measure too - pin PA7 (pin 7) except
  // on 14-pin DD-series, where it's PD7 (Pin 8)
  #if defined(SWAP_SERIAL)
    // That can be defined to swap serial pins.
    Serial.swap(1);
  #endif
  Serial.begin(115200);
  // Give the user a chance to connect to the serial port
  for (byte i = 3; i; i--) {
    digitalWrite(LED_BUILTIN, CHANGE);
    Serial.print('.'); // Show something so it doesn't look hung
    delay(1000);
  }
  digitalWrite(LED_BUILTIN, LOW);
  Demo();
  // Once we're done with the demo, we just sit here.
}

void Demo() {
  readHalfPage(BASE_ADDRESS);
  Serial.println(F("First, make sure we can write the flash"));
  byte check = Flash.checkWritable();
  if (check != FLASHWRITE_OK) {
    Serial.print(F("Error: "));
    Serial.println(Flash.checkWritable());
  }
  Serial.print(F("Erase the page: "));
  Serial.println(Flash.erasePage(BASE_ADDRESS, 1));
  readHalfPage(BASE_ADDRESS);
  Serial.println(F("Write some stuff to it:"));
  Serial.print(F("Word 0xCFF3 at base + 0: "));
```

```

Serial.println(Flash.writeWord(BASE_ADDRESS, 0xCFF3));
Serial.print(F("the number 1056 at base + 4: "));
Serial.println(Flash.writeWord(BASE_ADDRESS + 4, 1056));
//1056 is easy to see in hex, binary too: 0x0420, or 00000100 00100000
Serial.print(F("Word 0x8001 at base +7: "));
Serial.println(Flash.writeWord(BASE_ADDRESS + 7, 0x8001));
Serial.println(F(" Failed - words writes must be aligned"));
Serial.print(F("Byte 00 at base + 8: "));
Serial.println(Flash.writeByte(BASE_ADDRESS + 8, 0));
Serial.print(F("Byte 12 at base + 9: "));
Serial.println(Flash.writeByte(BASE_ADDRESS + 9, 0x12));
// Bytes do not need to be aligned.
Serial.print(F("3 words starting at base + 12: "));
Serial.println(Flash.writeWords(BASE_ADDRESS + 12, testArray, 3));
Serial.print(F("10-byte long array starting at base + 0x20: "));
Serial.println(Flash.writeBytes(BASE_ADDRESS + 0x20, testArray3, 10));
Serial.print(F("15-byte long character array at base + 0x40: "));
Serial.println(Flash.writeBytes(BASE_ADDRESS + 0x40, (uint8_t *)testArray2,
15));
Serial.println(F("After writing, flash looks like"));
readHalfPage(BASE_ADDRESS);
Serial.println(F("Read back the words we wrote at base + 12: "));
Serial.println(Flash.readWord(BASE_ADDRESS + 12), HEX);
Serial.println(Flash.readWord(BASE_ADDRESS + 14), HEX);
Serial.println(Flash.readWord(BASE_ADDRESS + 16), HEX);
Serial.println(F("Get a pointer to that string, cast it to char*, and print
it"));
uint8_t *ptr = Flash.mappedPointer(BASE_ADDRESS + 0x40);
if (ptr != NULL) {
    Serial.println((char *)ptr);
    Serial.print(F("No check for addresses being erased - write 0x2F to base +
0x43: "));
    Serial.println(Flash.writeByte(BASE_ADDRESS + 0x43, 0x18));
    Serial.println(F("And the string is now mangled:"));
    Serial.println((char *)ptr);
}
}

void loop() { /* do nothing */
}

// pretty-print 256b of flash
void readHalfPage(uint32_t StartAddress) {
    StartAddress &= 0x0001FF00;
    Serial.print(F("0x"));
    Serial.print(StartAddress, HEX);
    Serial.println(F(":"));
    for (unsigned long i = StartAddress; i < (StartAddress + 0x100); i++) {
        Serial.printHex(Flash.readByte(i)); //DxCore helper function - does the
leading 0.
        if ((i & 31) != 31) {
            //Print 32 bytes per line, space between them, (31 spaces, fence post)
            Serial.print(' ');

```

```
    } else {  
        // Last one of a line gets a newline printed after it instead.  
        Serial.println();  
    }  
}  
Serial.println();
```

Result

Example failed to compile for board AVR DA-series

Error Message

```
C:\Users\ivanFernandez\AppData\Local\Arduino15\packages\Microchip\hardware\megaavr  
\1.0.0\libraries\Flash\src\Flash.cpp:23:4: error: #error "You must also enable  
writing to flash from app in tools menu."  
    #error "You must also enable writing to flash from app in tools menu."  
    ^~~~~  
exit status 1  
Error compiling for board AVR DA-series.
```

Possible Fixes

1. Add an option to Tools menu to give the ability to enable writing to flash for the AVR DA-series boards.
2. Set a default writing to flash option within Boards.txt for the Microchip AVR128DA48 board.