

Code

```
// PROJECT   : EEPROM Burner
// AUTHOR    : R. Jamal (adapted from Ben Eater)
// PURPOSE   : To burn an EEPROM with data from an array
// COURSE    : ICS4U-E
// DATE      : 26 10 2024
// MCU       : MEGA328P (Nano)
// STATUS    : Working
#define EEPROM_D0 5 //Lowest EEPROM I/O pin
#define EEPROM_D7 12 //Highest EEPROM I/O pin
#define WRITE_EN PB5 //Write enable pin
#define LATCH PD3 //Shift register latch pin
#define DATA PD2 //Shift register serial pin
#define CLOCK PD4 //Shift register clock pin

byte data[] = {0x82, 0x10, 0x21, 0x62, 0xa0, 0xff, 0xff, 0xff, //Program codes
               0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
               0x05, 0x60, 0x07, 0x61, 0x80, 0x10, 0x81, 0x51,
               0xaa, 0xcd, 0x00, 0x62, 0xc0, 0x01, 0x62, 0xc1
               };

void setup() {
    Serial.begin(57600); //Begin serial comm
    DDRD |= 1 << DATA | 1 << CLOCK | 1 << LATCH; //Set SR pins as outputs
    DDRB |= 1 << WRITE_EN; //Set write enable as out
    PORTB |= 1 << WRITE_EN; //Pull write enable high

    Serial.print("Programming EEPROM"); //Signify start of process
    for (uint16_t address = 0; address < sizeof(data); address++) //Loop through
        writeEEPROM(address, data[address]); //Write each address
    printContents(); //Print EEPROM contents
}

void loop() {} //Nothing to do

void setAddress(int address, bool outputEnable) { //Sets address on SR
    PORTD &= ~(1 << LATCH); //Pull latch low
    ShiftPortD(DATA, CLOCK, MSBFIRST, (address >> 8) | (outputEnable ? 0x00 : 0x80));
    ShiftPortD(DATA, CLOCK, MSBFIRST, address); //Shift out second byte
    PORTD |= 1 << LATCH; //Pull latch high
}

void ShiftPortD(uint8_t dataPin, uint8_t clockPin, uint8_t bitOrder, uint8_t val) {
    for (uint8_t curBit = 0; curBit < 8; curBit++) {
        if (!bitOrder) { //LSBFIRST = 0
            if (val & (1 << curBit)) PORTD |= (1 << dataPin); //Set dataPin HIGH
            else PORTD &= ~(1 << dataPin); //Set dataPin LOW
        }
        else { //MSBFIRST = 1
            if (val & (1 << (7 - curBit))) PORTD |= (1 << dataPin); //Set dataPin HIGH
            else PORTD &= ~(1 << dataPin); //Set dataPin LOW
        }
        PORTD |= (1 << clockPin); //Set clockPin HIGH
        PORTD &= ~(1 << clockPin); //Set clockPin LOW
    }
}

byte readEEPROM(int address) { //Reads EEPROM at address
    DDRD &= ~(0b11100000); //Set I/O pins as inputs
    DDRB &= ~(0b00011111); //Set I/O pins as inputs
    setAddress(address, true); //Set address on SR
    byte data = 0; //Buffer for read data
    for (int pin = EEPROM_D7; pin >= EEPROM_D0; pin -= 1) //Loop through bits
        data = (data << 1) + digitalRead(pin); //Fetch each bit
    return data;
}
```