

Design Assignment 1

Student Name: Abraham Garcia

Student #: 5005262049

Student Email: garci11@unlv.nevada.edu

Primary Github address: https://github.com/SON-Abe/submission_da.git

Directory: C:/Users/abrah/OneDrive/Desktop/General/Uni/2024/CPE 301/Design_Assignments

Video Playlist:

Submit the following for all Labs:

1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

List of Components used

Block diagram with pins used in the Atmega3PB (only)

none

2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

none

3. **DEVELOPED/MODIFIED CODE OF TASK 2/A from TASK 1/A**

5.

```
.include <m328pdef.inc>
```

```
ldi r20, 0x0A ;r20 deincrement to stop when done reading my
hex
```

```
ldi ZH, HIGH(AVR<<1)           ;load AVR high byte << 1 into ZH
```

```
ldi ZL, LOW(AVR<<1) ;load AVR low byte << 1 into ZL
```

```
ldi XH, 0x01 ;high byte for x pointer address
```

```
ldi XL, 0x40 ;low byte for x pointer address
```

```
ldi YH, 0x02 ;high byte for y pointer address
```

```
ldi YL, 0x00 ;low byte for y pointer address
```

```
jmp SUM ;jump to SUM to add all hex in AVR
```

```
.org (0x1EEF>>1) ;sets origin to 0x1EEF right shift 1
```

AVR: .DW 0x4156, 0x5220, 0x6173, 0x7365, 0x6d62, 0x6c79, 0x2063, 0x6f64, 0x6520, 0x7468
;hex values

SUM:

```
cpi r20, 0          ;check if we finished reading all hex in AVR
breq STORE_IN_EEPROM ;jump to store in EEPROM when we finish

lpm r16, Z+         ;load in lower byte of hex, then increment
    lpm r17, Z+      ;load in higher byte of hex, then increment

add r18, r16        ;add lower bytes
    adc r19, r17      ;add higher bytes & lower byte overflow
    adc r23, r22      ;holds higher bytes overflow
    adc r25, r24      ;highest overflow

dec r20             ;dec r20 to let first line of SUM know if we finished

rjmp SUM            ;loop to to the top of SUM
```

STORE_IN_EEPROM:

```
SBIC EECR, EEPE     ;skips next line if EEPROM is currently writing
RJMP STORE_IN_EEPROM ;constantly loop till done writing

OUT EEARH, YH       ;sets Y pointer high byte to EEPROM address high byte
```

OUT EEARL, YL ;sets Y pointer low byte to EEPROM address low byte

OUT EEDR, r18 ;write r18 to EEPROM data register

SBI EECR, EEMPE ;enables EEPROM write

SBI EECR, EEPE ;starts EEPROM write

LD R21, Y+ ;increment Y pointer

STORE_IN_EEPROM2:

SBIC EECR, EEPE ;skips next line if EEPROM is currently writing

RJMP STORE_IN_EEPROM2 ;constantly loop till done writing

OUT EEARH, YH ;sets Y pointer high byte to EEPROM address high byte

OUT EEARL, YL ;sets Y pointer low byte to EEPROM address low byte

OUT EEDR, r19 ;write r19 to EEPROM data register

SBI EECR, EEMPE ;enables EEPROM write

SBI EECR, EEPE ;starts EEPROM write

LD R21, Y+ ;increment Y pointer

STORE_IN_EEPROM3:

SBIC EECR, EEPE ;skips next line if EEPROM is currently writing

RJMP STORE_IN_EEPROM3 ;constantly loop till done writing

OUT EEARH, YH ;sets Y pointer high byte to EEPROM address high byte
OUT EEARL, YL ;sets Y pointer low byte to EEPROM address low byte
OUT EEDR, r23 ;write r23 to EEPROM data register

SBI EECR, EEMPE ;enables EEPROM write
SBI EECR, EEPE ;starts EEPROM write

LD R21, Y+ ;increment Y pointer

STORE_IN_EEPROM4:

SBIC EECR, EEPE ;skips next line if EEPROM is currently writing
RJMP STORE_IN_EEPROM4 ;constantly loop till done writing

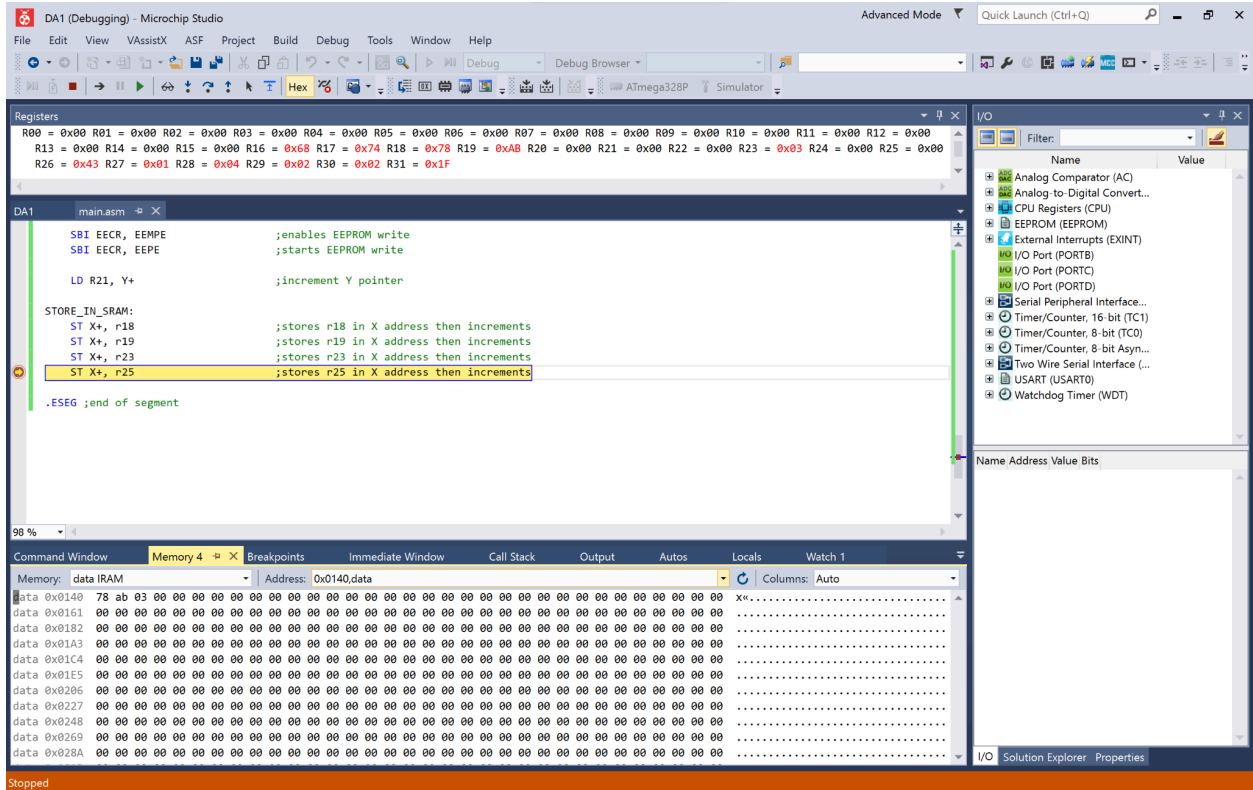
OUT EEARH, YH ;sets Y pointer high byte to EEPROM address high byte
OUT EEARL, YL ;sets Y pointer low byte to EEPROM address low byte
OUT EEDR, r25 ;write r25 to EEPROM data register

SBI EECR, EEMPE ;enables EEPROM write
SBI EECR, EEPE ;starts EEPROM write

LD R21, Y+ ;increment Y pointer

STORE_IN_SRAM:

Task 1:



7. VIDEO LINKS OF EACH DEMO

8. GITHUB LINK OF THIS DA

Task 1:https://github.com/SON-Abe/submission_da.git

Student Academic Misconduct Policy

<http://studentconduct.unlv.edu/misconduct/policy.html>

"This assignment submission is my own, original work".

Abraham Garcia