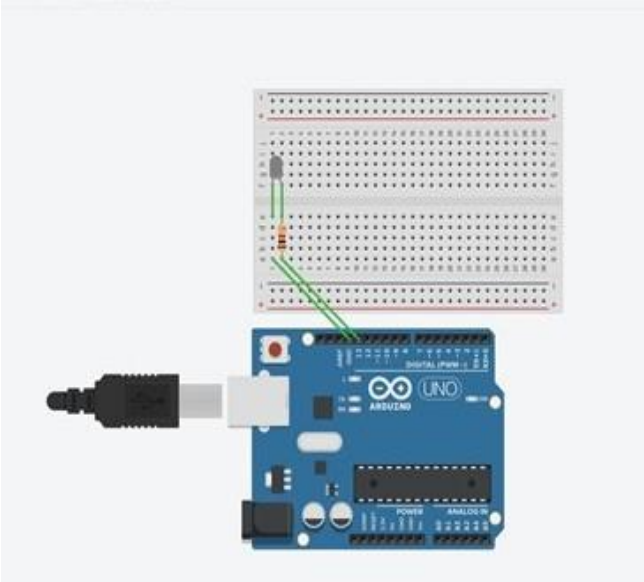
LAB REPORT #4

(Names) Kenilkumar Patel

(Date) 4/8/2021

Screenshot and components



Memory Mapped I/O - Memory Spaces and I/O are in the same addresses in the RAM.

Port Mapped I/O - Memory Spaces and I/O are in different addresses spaces as I/O is on a different chip.

Process:

First, in the setup, I called displayRam to show the hex data bits from 0x20 to 0x2f since the address for pins 2 to 13 are in there by calling displayRAM((char *) 0x20, (char *) 0x2f, true). Then I called pinMode(13, OUTPUT) to turn on pin 13 in order to see the changes in any of the hex bits when I call displayRam again. I saw a change at address 0x24 which gives a hex value of 0x20. I then call displayBits((char *) 0x24) which gives the binary form of hex 0x20 which is

100000. To turn off the pin 13, I used pinMode(13, INPUT). Then, I called displayRAM again and the hex data bit at 0x24 changed back to 00. This fully shows that Pin 13's address is at 0x24 and the data bit is 0x20. I also learned how to check the RAM memory of the pins using methods described in the lab. Finally, I learned how to flash a light directly without using digitalWrite().

Conclusion:

I learned that bits are organized by purposes not by pins. I learned that each pin has 3 bits ,one to write to, one to control the direction of the data, and one where we receive the input from.

Pin (number)	Address (hex)	Data bit (hex)
2	2a	04
3	2a	08
4	2a	10
5	2a	20
6	2a	40
7	2a	80
8	24	01

9	24	02
10	24	04
11	24	08
12	24	10
13	24	20

```
/*
  Memory Mapped I/O Exploration
  */

/* Figure out what bits control pin 13. make use of the provided
functions *  to do this. */
void setup() {
  // Setup for Serial output
  Serial.begin(9600);

  displayRAM((char *) 0x20, (char *) 0x2f, true);
  pinMode(13, OUTPUT);
  displayRAM((char *) 0x20, (char *) 0x2f, true);
  //displayBits((char *) 0x24);
  //pinMode(13, INPUT);
  // displayRAM((char *) 0x20, (char *) 0x2f, true);
  // displayBits((char *) 0x24); char
  *example = (char *) 0x24; int theBit =
  (*example >> 5) & 1;
  Serial.println(theBit, BIN);



}

/*once you know which bit can be used to turn pin 13 on and off,
* try to blink an led in loop without using digitalWrite().
*/ void
loop() {
  pinMode(13, OUTPUT);
  char *pin13 = (char *) 0x24;
  *pin13 ^= 0b100000;
  //delay(1000);
```

```
}

/* example call displayRAM((char *) 0x8E0, (char *) 0x8FF, true);
* if hex is false, letters and numbers will be printed, and all other values will be represented as '.' */
void displayRAM(char *start, char *endd, bool hex) {
 char *array;
 for(array = start; array < endd; array += 0x10) {
   //create row number if
   (array < (char *)0x10)
     Serial.print('0'); if (array
   < (char *)0x100)
   Serial.print('0');
   Serial.print((int)array, HEX);
   Serial.print(": ");
   //for each index (0 through 15 inclusive)
   for(int i = 0; i < 0x10; i++) {
     if(hex) {
       if (array[i] >= 0x00 && array[i] < 0x10)
         Serial.print('0');
       Serial.print(array[i] & 0xFF, HEX); //0xFF is our bitmask
     } else {
       Serial.print((array[i] >= ' ' && array[i] <= 'z') ? array[i] : '.');
     }
     Serial.write(' ');
   }
   Serial.println();
 }
 Serial.println();
}

//Example call displayBits((char *)
0x100); //pretty prints an address in binary
void displayBits(char *address) {
Serial.print("0x");
 Serial.print((int) address, HEX);
 Serial.print(": ");
  Serial.println(address[0], BIN);
}
```