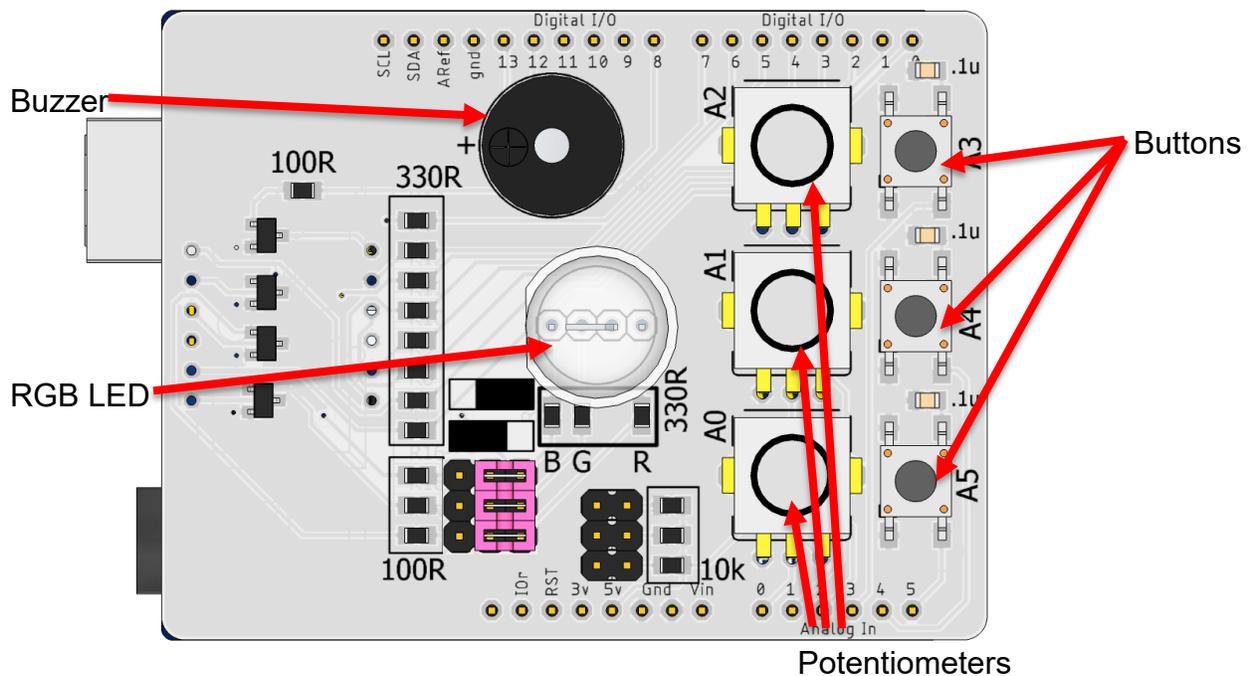


For the next several labs you will be using an Arduino UNO board with a Atmega 328 microcontroller, and a shield (piggyback board) custom designed for this course.

NOTE: although we are using the Arduino environment, unless specifically allowed, you may not use any Arduino type "easy" routines, specifically `digitalRead()` and `digitalWrite()` are prohibited.

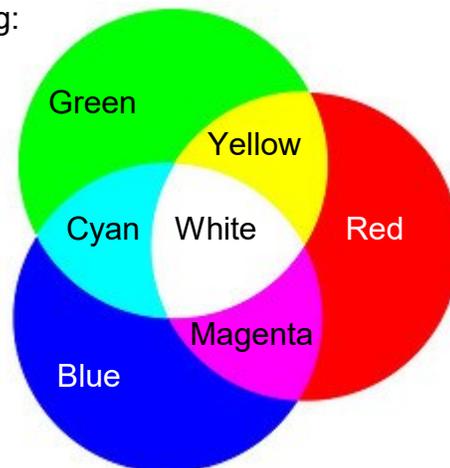
The board you are using looks as follows:



For this lab, you are going to focus solely on the RGB LED. Your task for this lab is to make the LED change through the following color sequence and repeat continuously. Each color should hold for about 1 second.

Red → Yellow → Green → Cyan → Blue → Magenta, and repeat.

This LED is a tri-color LED, meaning it has a red, a green, and a blue LED encompassed in one piece of plastic. To create other colors from these three, you can use the following rules of color mixing:



In chart form, the above color diagram may be represented as:

Red bit	Green bit	Blue bit	Color
0	0	0	Black(off)
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta (Purple)
1	1	0	Yellow (Amber)
1	1	1	White

PORTS:

Ports are special registers in memory that relate to the IO pins. There are 3 ports on the Arduino Uno, B, C, and D. Port D is pins 0-7. Port B is pins 8-13. Port C is the analog input pins A0-A5. Each port has 3 registers associated with it.

DDRx	Data Direction Register. This register is used to specify if a particular bit of PORTx is input or output. On any given port, some bits may be output and some may be input.
PORTX	Used to output to PORTx. May also be used to get the current state of the bits of a port which are already set to output.
PINx	Used for reading from PORTx

To set the output of PORTB bit 1 to low, you would use the following code:

```
// set portB direction for bit 1 to output
// note no other bits are changed in DDRB
DDRB = DDRB | 0b00000010;    // Could use 1<<1
// set bit 1 or PORTB to 0 (low)
PORTB = PORTB & 0b11111101;  // Could use ~(1<<1)
```

To read the input from portC bit 6, you would use the following code:

```
// insure bit 6 or PORTC is set for input
DDRC = DDRC & 0b10111111; // could use ~(1<<6)
x = PINC // read value of PINC
// code to mask for the bit you want
```

On the dev board the RGB LED is connected to port B.

Segment	Port B bit	Value to write to turn on
Common	b4	1
Red	b3	0
Green	b2	0
Blue	b1	0

Write a program that loops through all the colors in the table above and displays them on the RGB LED. Each color should be displayed for one second and the program should loop forever. Be sure to add the appropriate documentation and submit the .ino file to the M drive.

Arduino Environment

Unlike c in Visual Studio, the Arduino environment is a little different. Instead of a "main()" function, there is a setup() function and a loop() function.

The setup() function is done once. The loop() function is repeated forever.

```
void setup()
{
    // code to be done once
    // (such as setting DDRx)
}

void loop()
{
    // code to be repeated
}
}
```

This must be checked of by TA, AND submitted to your M: drive

Student Name: _____

TA Signature: _____

Date/Time: _____

Checksum: _____