Happy 12th Birthday Arduino!

LAS POSITAS COLLEGE

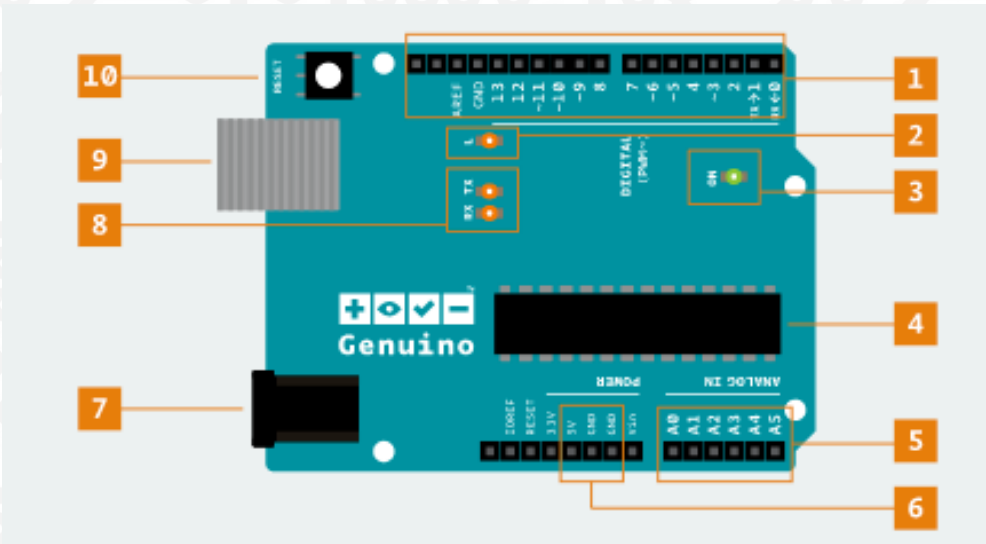
ARDUINO DAY 2016

Saturday, April 2 at Robot Garden, 11AM to 3PM

Robot Garden

day.arduino.cc

Anatomy of an Arduino



1. Digital pins
2. Pin 13 LED
3. Power LED
4. ATmega microcontroller
5. Analog in
6. GND and 5V pins
7. Power connector
8. TX & RX LEDs
9. USB port
10. Reset button

Arduino UNO is a minimal ecosystem to support the microcontroller

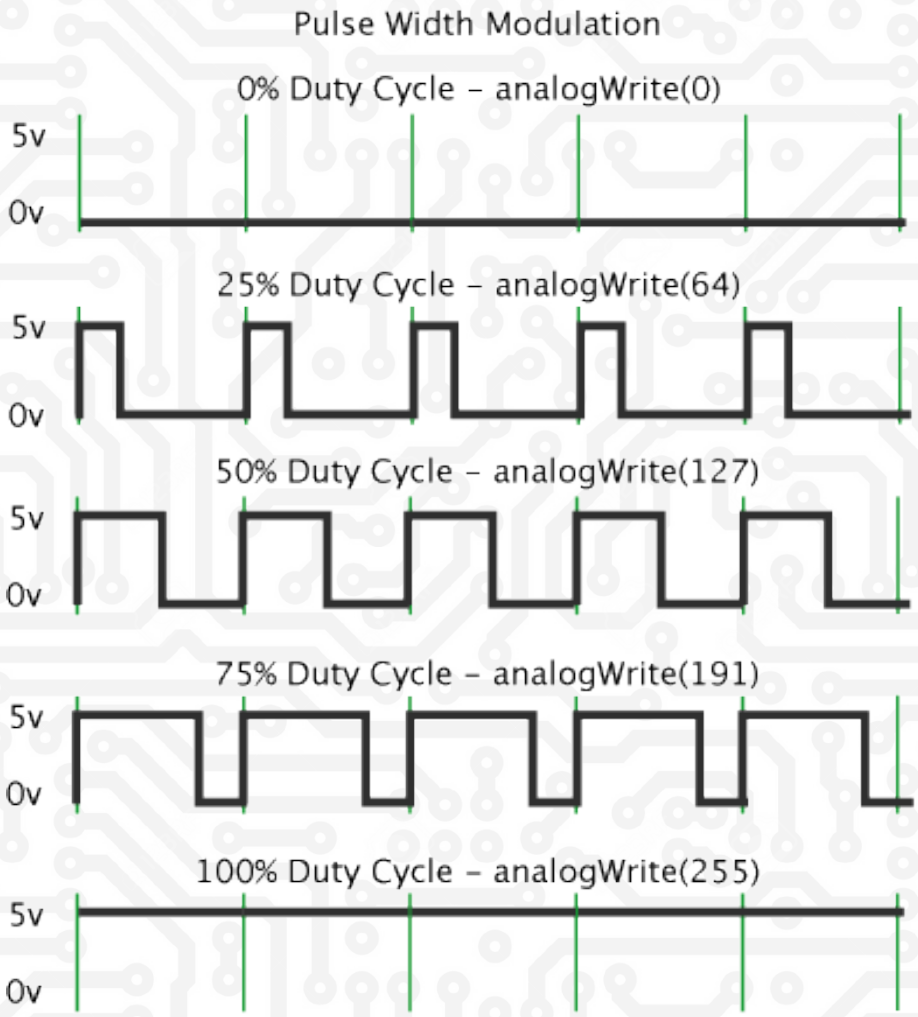
- ATMega328 microcontroller
  - One of the first chip families w/ onboard flash memory
  - 32 kbytes of memory
  - 8-bit architecture.
  - Added for Arduino: boot loader (~0.5 kbyte)

- USB to serial interface
- 16 MHz crystal clock
- 7805 5 Volt voltage regulator

Certain Functions for Certain Pins

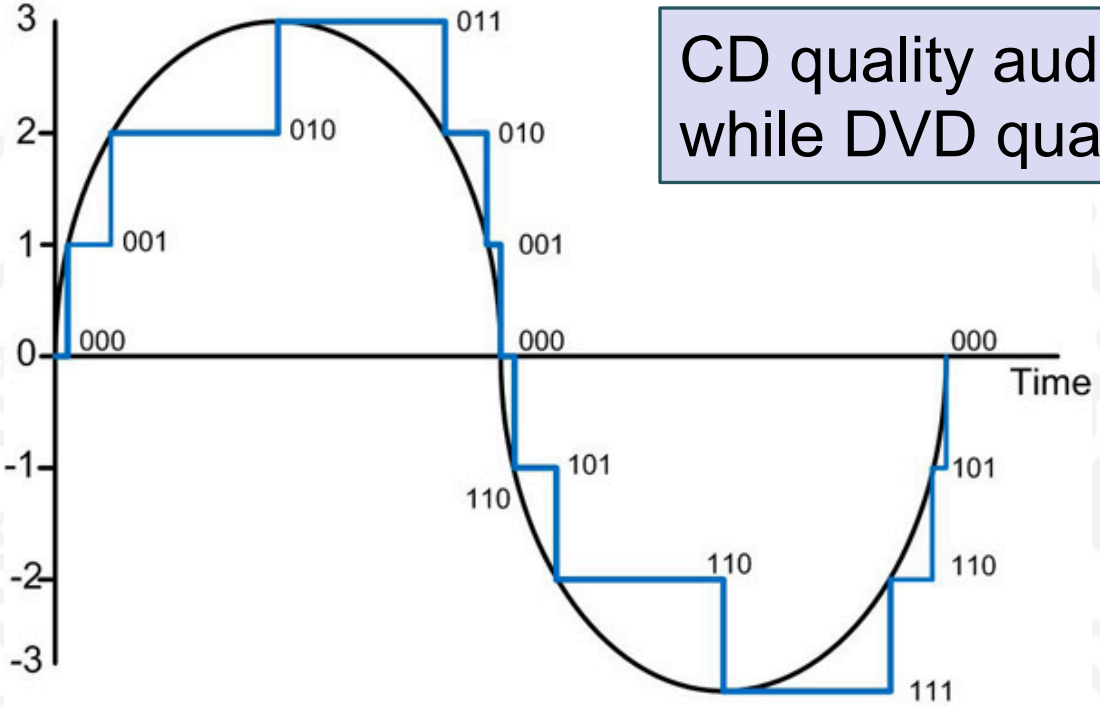
- Digital pins:
  - digitalRead(), digitalWrite()
  - w/ PWM symbol analogWrite()
- Analog pins:
  - analogRead() – 8-bit analog-to-digital converter (ADC)

Pulse Wave Modulation: PWM



The analogWrite() command mocks up an analog signal value between 0 and 255 by varying duty cycle. $2^8 = 256$ so this is 8-bit resolution.

Analog to Digital Conversion (ADC)



CD quality audio is 16-bit while DVD quality is 24-bit

Blue is a 3-bit ($2^3 = 8$) digital representation of the black analog sine wave

Arduino ADC pins are 10 bit, so they return values in the range from 0 to $2^{10} = 1024$

Light-chasing robot sketch: initialization

```
// pin assignments
int LDR[2] = {A1, A0};
int motorPin1[2] = {5, 3};
int motorPin2[2] = {6, 4};

// allocate variables:
int v[2];
int base[2];
int thresh[2] = {0, 0};
int drv[2];
int min[2]    = { 1000,  1000};
int max[2]    = {-1000, -1000};
```

Photoresistor2_to_HbridgeL293D.ino

Light-chasing robot sketch: setup()

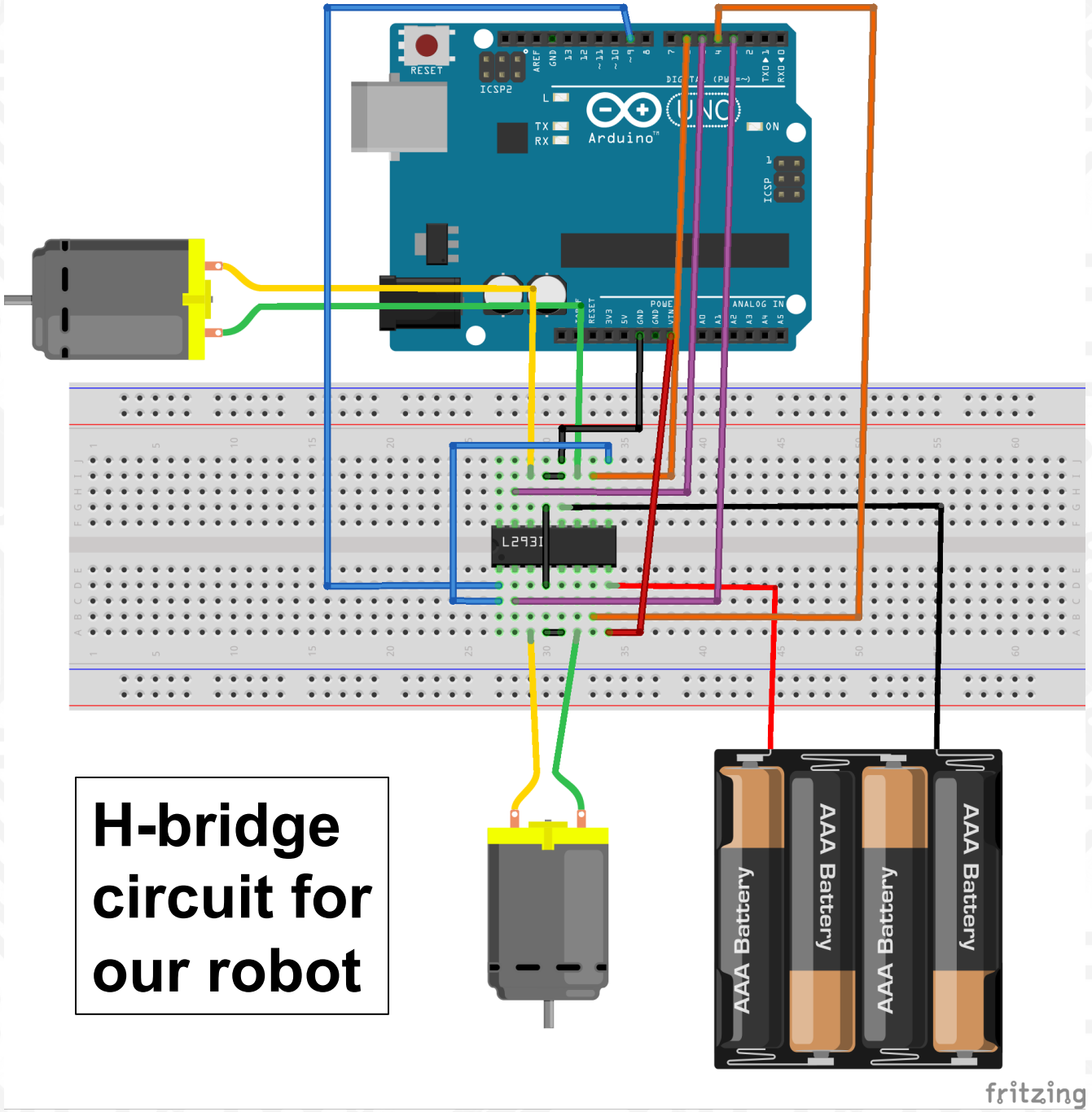
```
void setup() {
    // set up the pins:
    for (int i = 0; i <= 1; i++)
      {
       pinMode(LDR[i], INPUT);
       int base = analogRead(LDR[i]);
       min[i] = base;
       max[i] = base+100;
       pinMode(motorPin1[i], OUTPUT);
       pinMode(motorPin2[i], OUTPUT);
       analogWrite(motorPin1[i],0);
       analogWrite(motorPin2[i],0);
      }
}
```

Photoresistor2_to_HbridgeL293D.ino

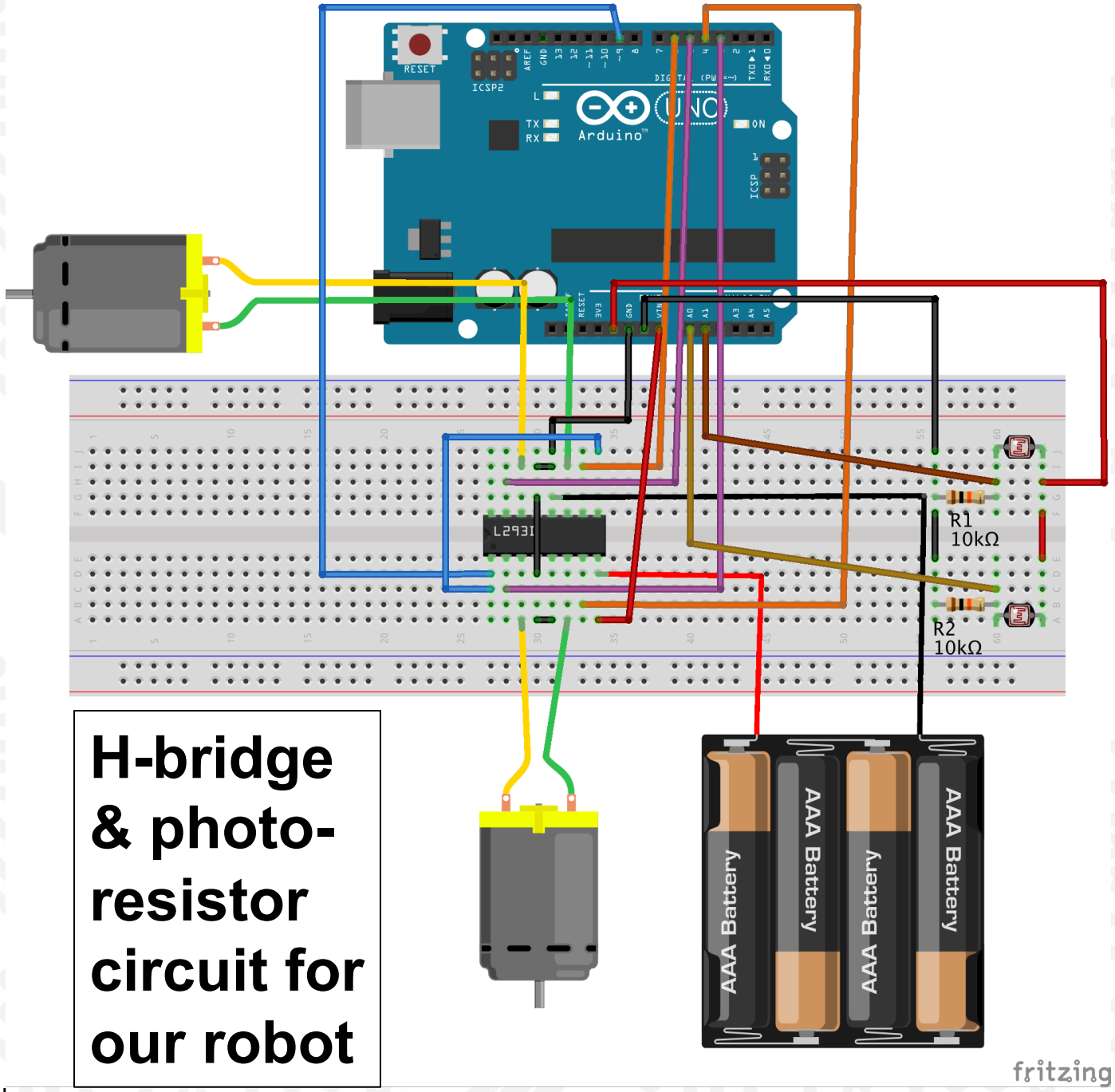
Light-chasing robot sketch: loop()

```
void loop() {

      // for each photoresistor 'i':
      for (int i = 0; i <= 1; i++) {
        // read the voltage of the photoresistor:
        v[i] = analogRead(LDR[i]);

        // find min and max value of the photoresistor:
        if (v[i] > max[i]) max[i] = v[i];
        if (v[i] < min[i]) min[i] = v[i];

        // Set the threshold to activate the motor at 35% between min and max
        int mx_mn35 = int(0.35*(max[i] - min[i]));
        thresh[i] = mx_mn35 + min[i];

        // If we're above threshold, then scale the motor voltage according to how far above threshold we are.
        if (v[i] > thresh[i]) {
          // minimum speed is 175.
          // maximum speed is 175 + 80 = 255.
          drv[i] = 175 + int( 80.0*( float(v[i] - thresh[i]) / (max[i] - thresh[i]) ));
        } else {
          drv[i] = 0; // remember to turn the motor off if below threshold
        }

        // send the voltage to the motor pins on the H-bridge:
        analogWrite(motorPin1[i], drv[i] );
        analogWrite(motorPin2[i], 0 );
      }

     // If either motor is driving, light the LED:
     if (drv[0] > 0 || drv[1] > 0) {
       digitalWrite(ledPin, HIGH);
     } else {
       digitalWrite(ledPin, LOW);
     }

}
```

Photoresistor2_to_HbridgeL293D.ino

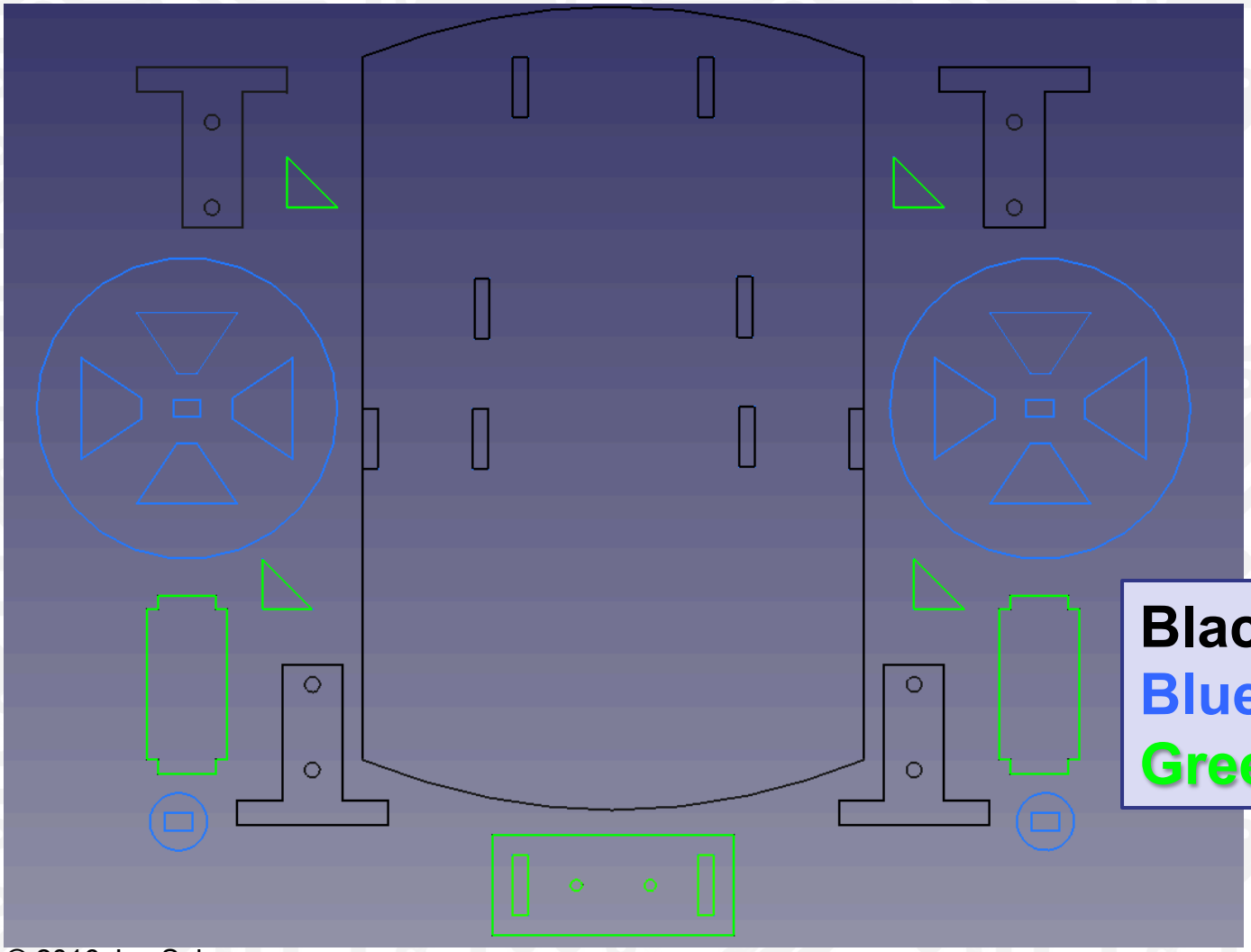


H-bridge circuit for our robot



H-bridge & photo-resistor circuit for our robot

Robot1c plans



Black – chassis
Blue – wheels
Green – nose gear

Units

Distance – meters (m)
Mass – kilograms (kg)
Time – seconds (s)

Charge – Coulomb (C)
Current – Amp (A)
Resistance – Ohm (Ω)
Magnetic Field – Tesla (T)

f – femto (10^{-15})
p – pico (10^{-12})
n – nano (10^{-9})
μ – micro (10^{-6})
m – milli (10^{-3})

k – kilo (10^{3})
M – mega (10^{6})
G – giga (10^{9})
T – tera (10^{12})
P – peta (10^{15})