



DAVID GARRETT

Portfolio



2/6/19

Perfect Attendance



To the Family and Friends of David Garrett,

It is my privilege to announce to you that David has achieved perfect attendance at Allied Moulded Products for the entire year of 2018. Paul and I understand that it takes not only the individual, but entire families, a great group of friends, or even an entire community to help your loved one make it to work every day on time. Our Allied family depends on David every single day and we understand that it's not always easy or convenient to balance the demands of everyday life and work.

Paul and I would like to thank you from the bottom of our hearts for all the support you have given to David. Spending time with loved ones has always been very important to me, Paul, and our family. Please accept our apologies for late nights, missed dinners, anniversary's, birthday's, sporting events, long business trips, sleep deprivation, and overtime hours. Please know these sacrifices do not go unnoticed and we are forever grateful to you and your family.

Allied wouldn't be what we are today without all the love and support of family and friends like you!

We are indeed grateful for the efforts and sacrifices that David made to accomplish perfect attendance.

Thank you,

Aaron

Career Readiness

CAREER READINESS CERTIFICATE

OF
ACCOMPLISHMENT
PROUDLY PRESENTED TO

David Garrett

In recognition of documenting all 10 competencies in the NSCC Career
Connections Readiness Badging Program

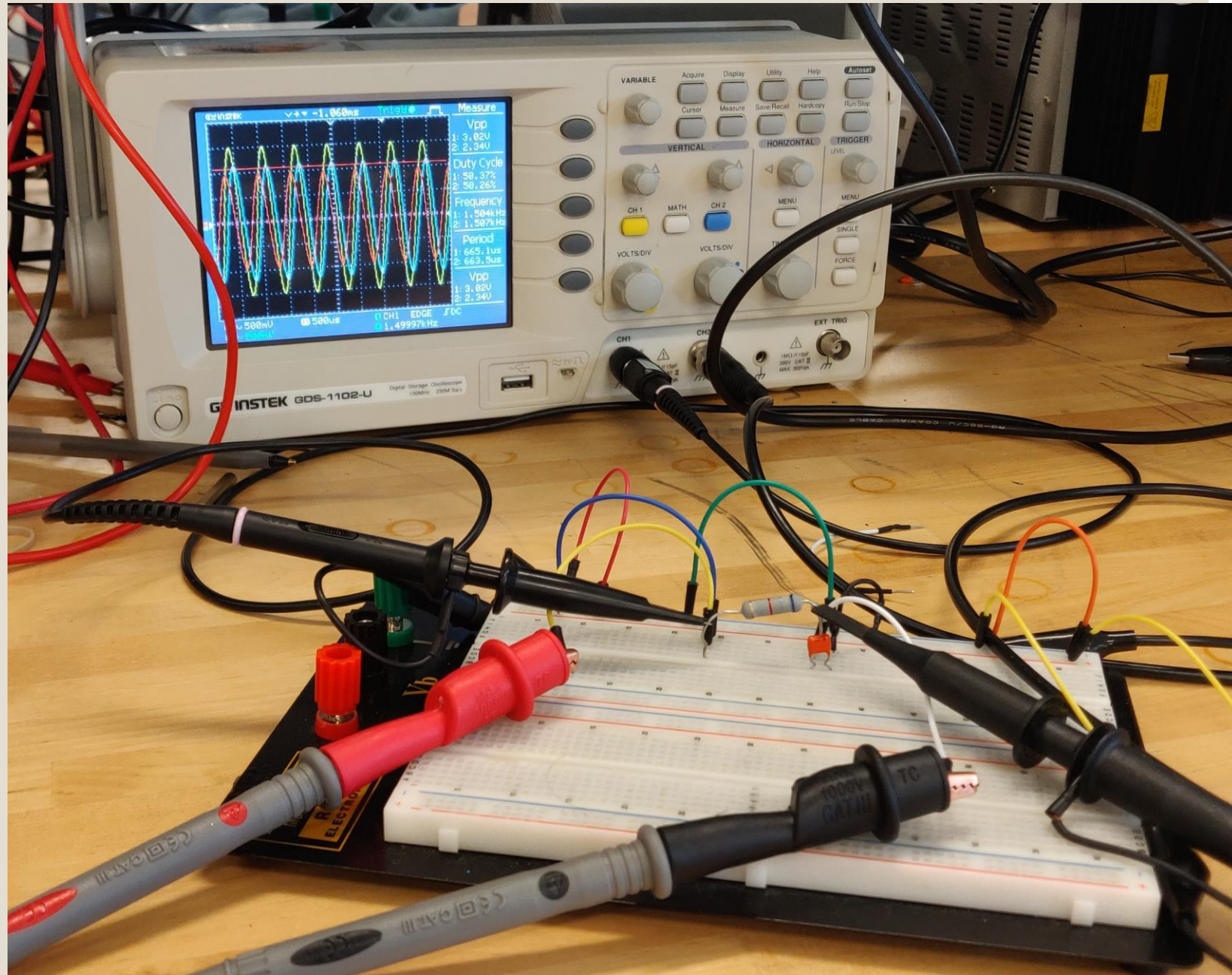
Brad Geer
BRAD GEER

11/17/2022
DATE

 Northwest State
Community College



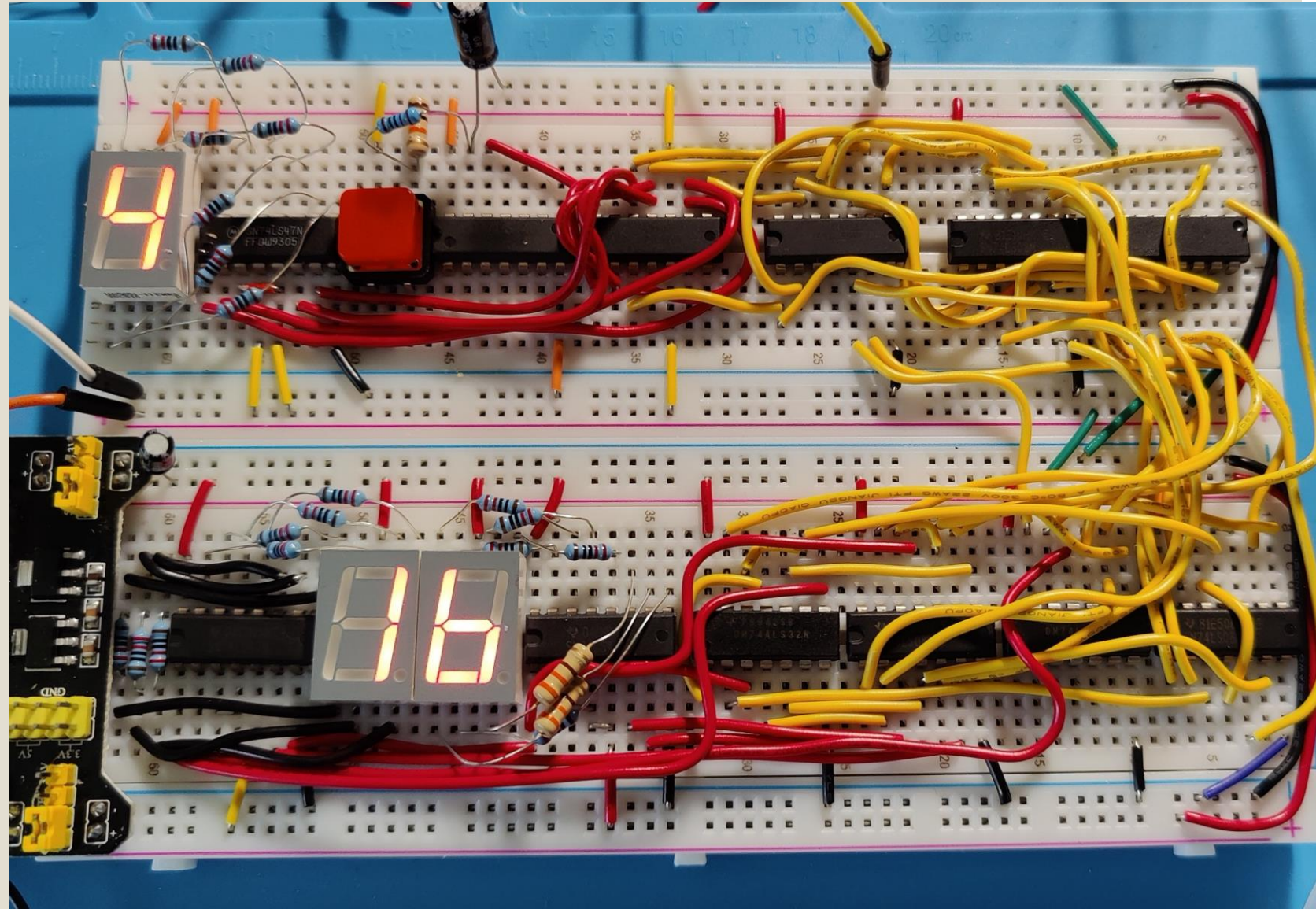
Oscilloscope



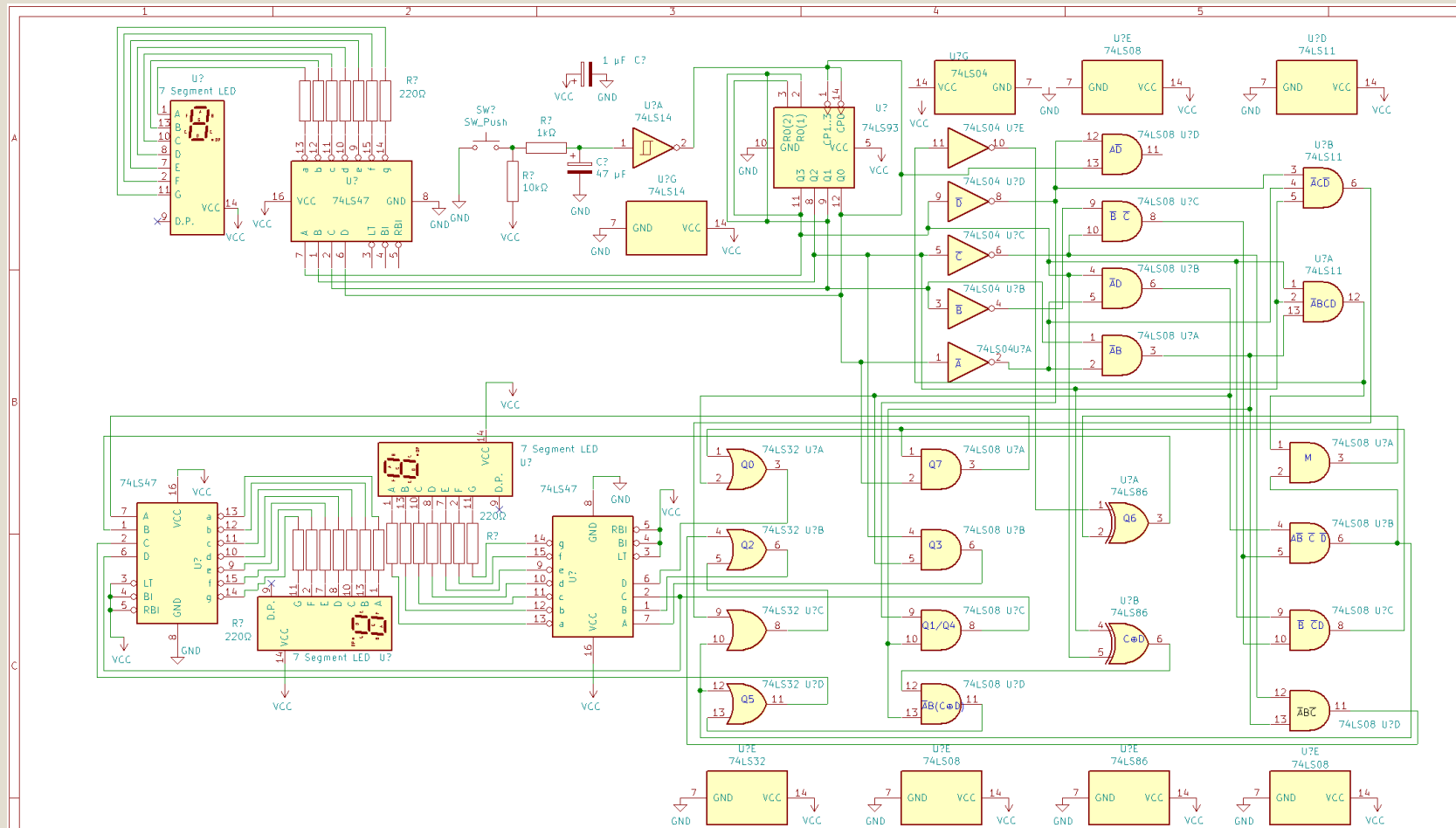
Digital Circuit

Function

- Display a number and the square of that number (Ex. 9 and 81)
- Cycle through each number to be squared using a push button
- Uses digital logic and integrated circuits to perform mathematical operations



Digital Circuit Schematic



Digital Circuit Design

Q_0

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	1	0
$\bar{A}B$	0	1	1	0
AB	x	x	x	x
$A\bar{B}$	0	1	1	x

Q_3

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	0
$\bar{A}B$	0	0	1	0
AB	x	x	x	x
$A\bar{B}$	0	0	x	x

$$Q_0 = \bar{A}D + \bar{B}\bar{C}D$$

Q_1

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	0	0	1
AB	x	x	x	x
$A\bar{B}$	0	0	x	x

$$Q_1 = \bar{A}B\bar{D}$$

Q_2

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	1
$\bar{A}B$	1	1	0	1
AB	x	x	x	x
$A\bar{B}$	1	0	x	x

$$Q_2 = \bar{A}B\bar{C} + \bar{A}C\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}(B\bar{C} + C\bar{D}) + \bar{A}B\bar{C}\bar{D}$$

$$Q_3 = \bar{A}C\bar{D}$$

Q_4

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	0	0	1
AB	x	x	x	x
$A\bar{B}$	0	0	x	x

$$Q_4 = \bar{A}B\bar{D}$$

Q_6

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	1	0
AB	x	x	x	x
$A\bar{B}$	1	0	x	x

$$Q_6 = \bar{A}B\bar{C}D + \bar{A}B\bar{C}\bar{D}$$

Q_5

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	1	1	0
AB	x	x	x	x
$A\bar{B}$	1	0	x	x

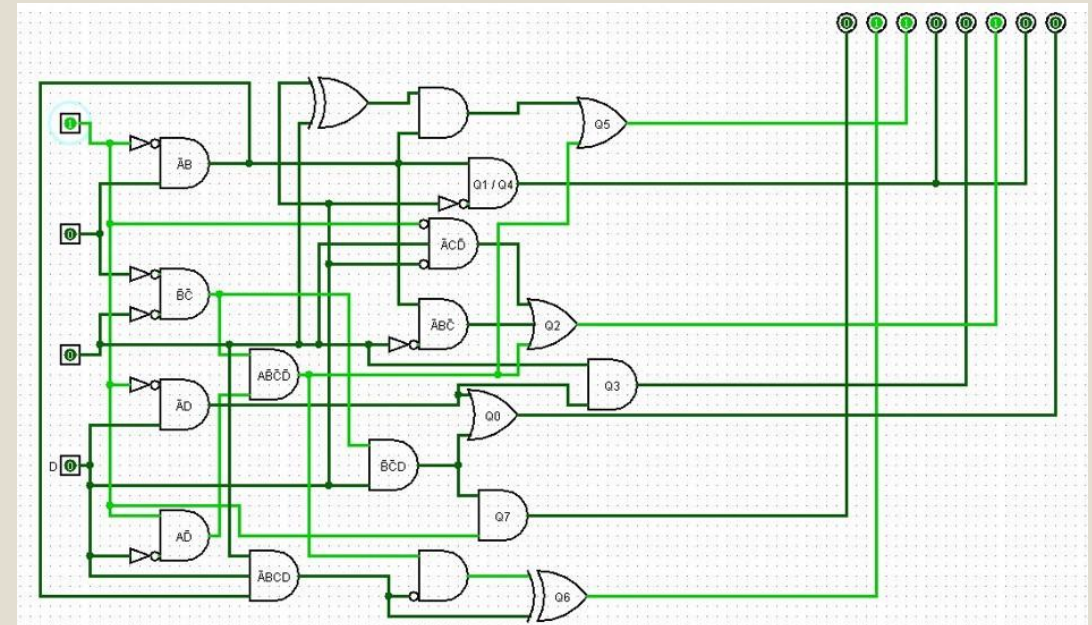
$$Q_5 = \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}B\bar{C}\bar{D}$$

Q_7

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	0
$\bar{A}B$	0	0	1	0
AB	x	x	x	x
$A\bar{B}$	0	1	x	x

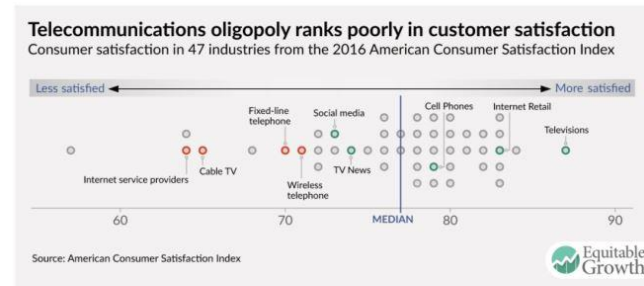
$$Q_7 = \bar{A}B\bar{C}\bar{D}$$

Counter Value	BCD				Square Value	BCD 2				BCD 1			
	A	B	C	D		Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0	0	0	0	1
2	0	0	1	0	4	0	0	0	0	0	1	0	0
3	0	0	1	1	9	0	0	0	0	1	0	0	1
4	0	1	0	0	16	0	0	0	1	0	1	1	0
5	0	1	0	1	25	0	0	1	0	0	1	0	1
6	0	1	1	0	36	0	0	1	1	0	1	1	0
7	0	1	1	1	49	0	1	0	0	1	0	0	1
8	1	0	0	0	64	0	1	1	0	0	1	0	0
9	1	0	0	1	81	1	0	0	0	0	0	0	1



Group Research Paper

Figure 9



Note. A communications oligopoly on steroids why antitrust enforcement and regulatory oversight in digital communications matter pg 17 telecoms customer satisfaction (Cooper & Kimmelman 2017).

Since the ISPs have a large captured customer base, then the ISP will use their leverage to keep prices high.

Some examples of lack of competition keeping prices high are:

- The Wall Street Journal did an analysis of 3300 different broadband bills and came to the conclusion that prices were higher in areas of less competition (Pacheco & Ramachandran 2019).
- Spectrum, a major broadband ISP, charges \$200 installation fee in areas without competition while only \$50 in areas with competition (Dampier 2021a)
- The price of Verizon and AT&T's mobile data remaining the same over the past three years despite the cost per bit declining by 40% per year since 2018. U.S. prices are double compared to Europe and quadruple to the world average. (Dampier 2021b).

Currently, the average price of wired broadband internet is

- DSL - \$50/mo.
- Cable - \$52/mo.
- Fiber - \$59/mo. (McNally 2020).

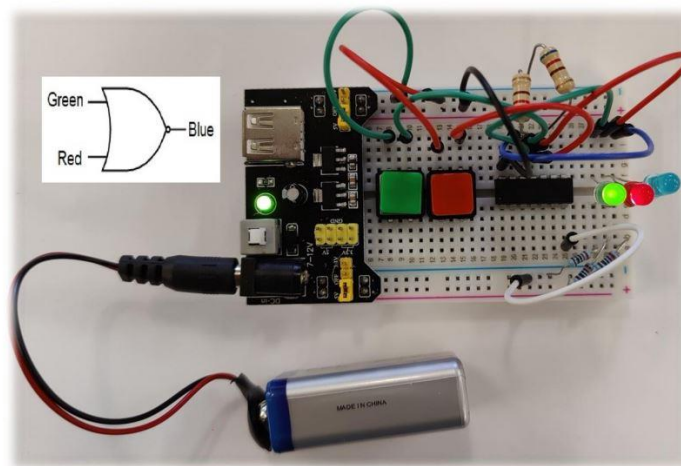
Those kinds of prices are out of reach for many families that could be helped with an increased amount of businesses competing against each other.

Comments from Instructor:

This was a great looking and well researched and delivered report. It is the absolute best report ever handed to me. The presentation itself worked on every level and by the group evaluation/survey demonstrated that even if not every group member was in class during the process, they did contribute toward the most excellent product. This report ticked every single box including the very workable solution... genius. The process getting to the final product was a bit more chaotic compared to the other group, however, this group seemed to feed off it and once the direction was focused and the assignments apportioned, the report took off. It was hampered by absences and radio silence by some members of this group but they seemed to have come through at the end. I cannot say enough about the members who showed up and worked the problem and set the stage for the success of the entire group in delivering an impeccable final product. **Note: several of the evaluations mentioned how much they appreciated you being in this group. nice!**

Instruction Paper

Creating a NOR gate circuit



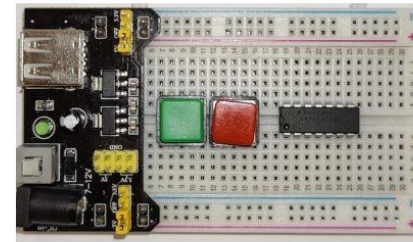
This manual explains how to set up a circuit that utilizes a 74LS02 integrated circuit or IC to show a NOR gate operation. The 74LS02 IC detects the voltage level of two inputs and outputs a voltage level based on the inputs. Here's a table of configuration of inputs of a NOR gate.

Green Input	Red Input	Blue Output
HIGH Voltage	HIGH Voltage	LOW Voltage
HIGH Voltage	LOW Voltage	LOW Voltage
LOW Voltage	HIGH Voltage	LOW Voltage
LOW Voltage	LOW Voltage	HIGH Voltage

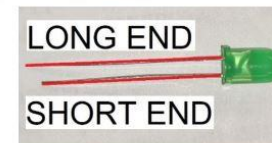
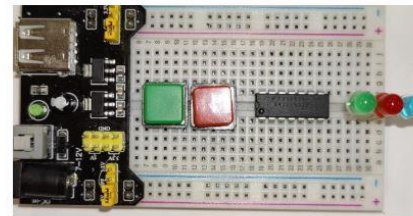
The circuit uses two button inputs to put a high or low voltage at the inputs of the 74LS02 IC. The circuit also has three LEDs to show the state of the inputs and outputs of the 74LS02 integrated circuit. HIGH means LED is on, LOW means LED is off.

Directions

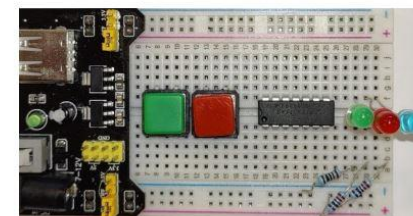
4. Insert the 74LS02 IC pins into column 19 through 25
 - Notch pointed towards left side of circuit
 - Seat IC gently. Pins break easily
 - Seat IC over the gap between row E and F



5. Insert 3 LEDs on right side of breadboard
 - Green LED Long End
 - Row F Column 28
 - Green LED Short End
 - Row E Column 28
 - Red LED Long End
 - Row F Column 29
 - Red LED Short End
 - Row E Column 29
 - Blue LED Long End
 - Row F Column 30
 - Blue LED Short End
 - Row E Column 30



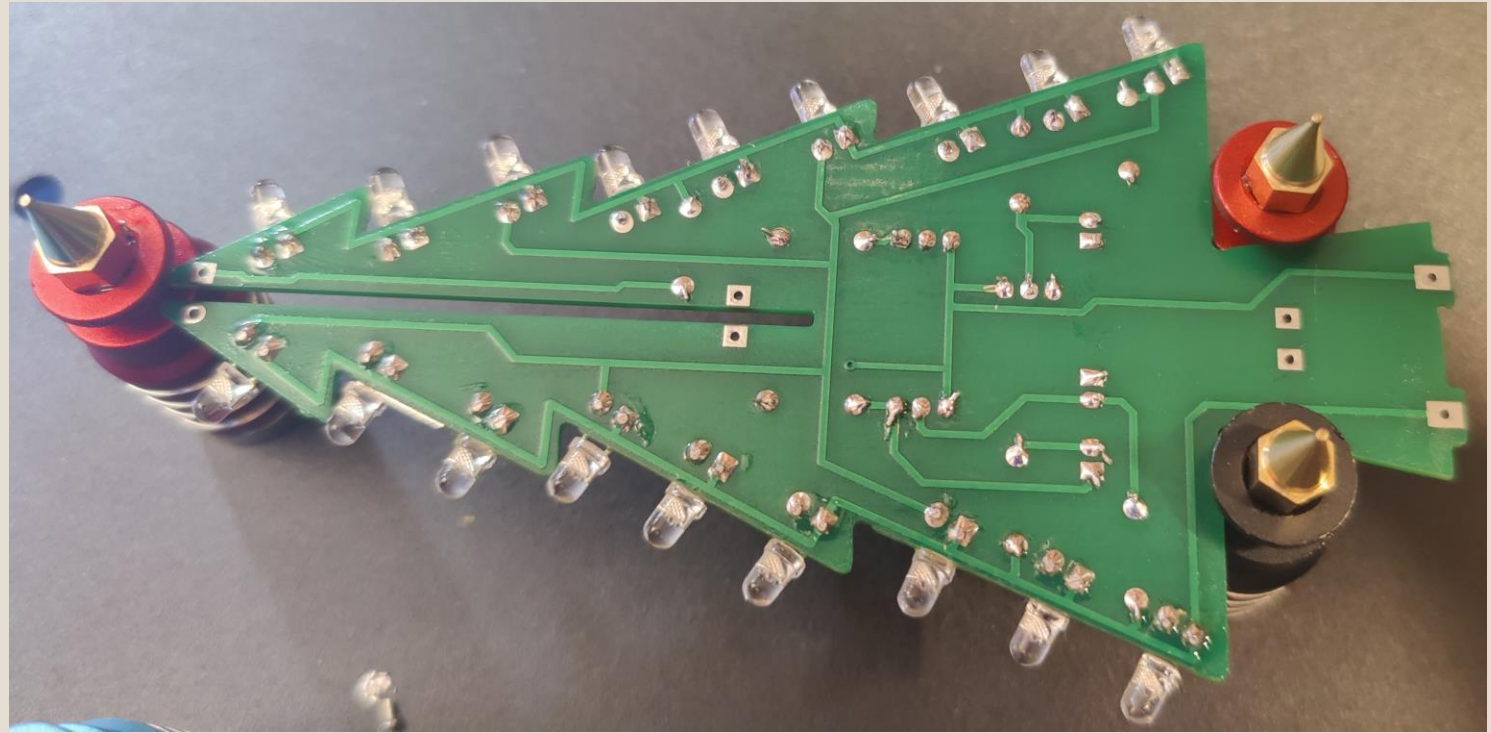
6. Insert 3 blue 330 Ω resistors with one side in the same column as the short end of each LED and the other side in the negative bus
 - Does not matter where on negative bus the resistors are placed



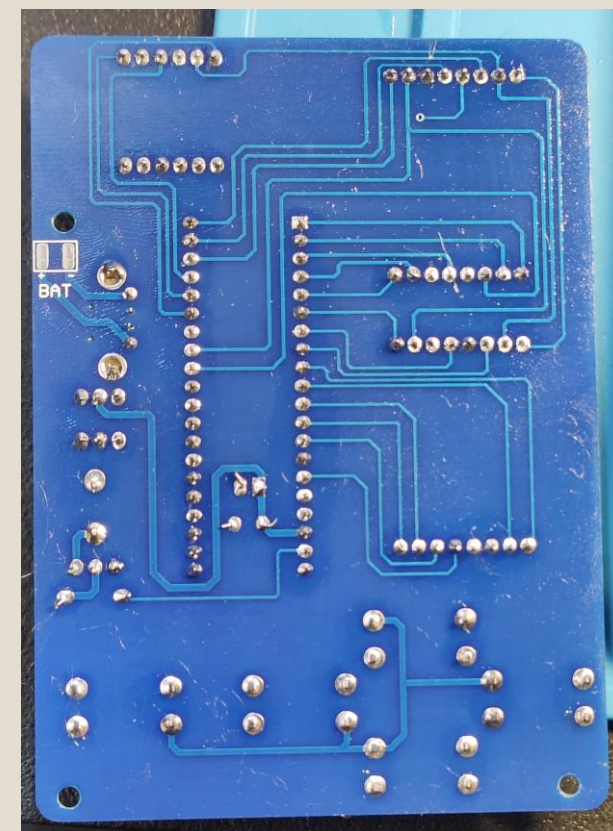
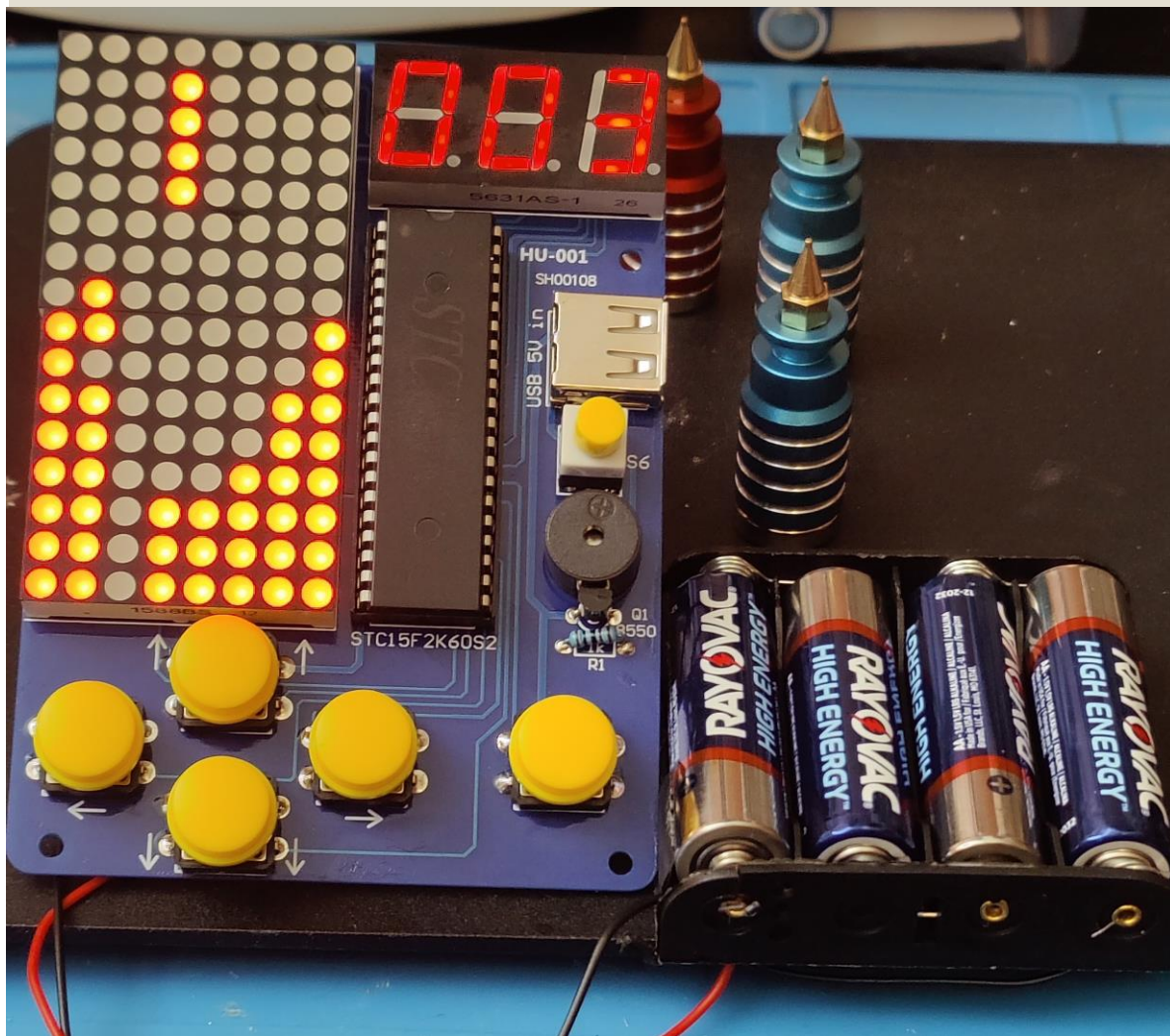
Comments from Instructor:

David, This was really saving the best for last. You took us through the process and then demonstrated it brilliantly including the lighting of the board. Paperwork and effort were both perfection. You make my job as instructor pretty easy... thanks for the continued effort in this class from day one. This was an outstanding effort!! Thanks.

Soldering



Soldering

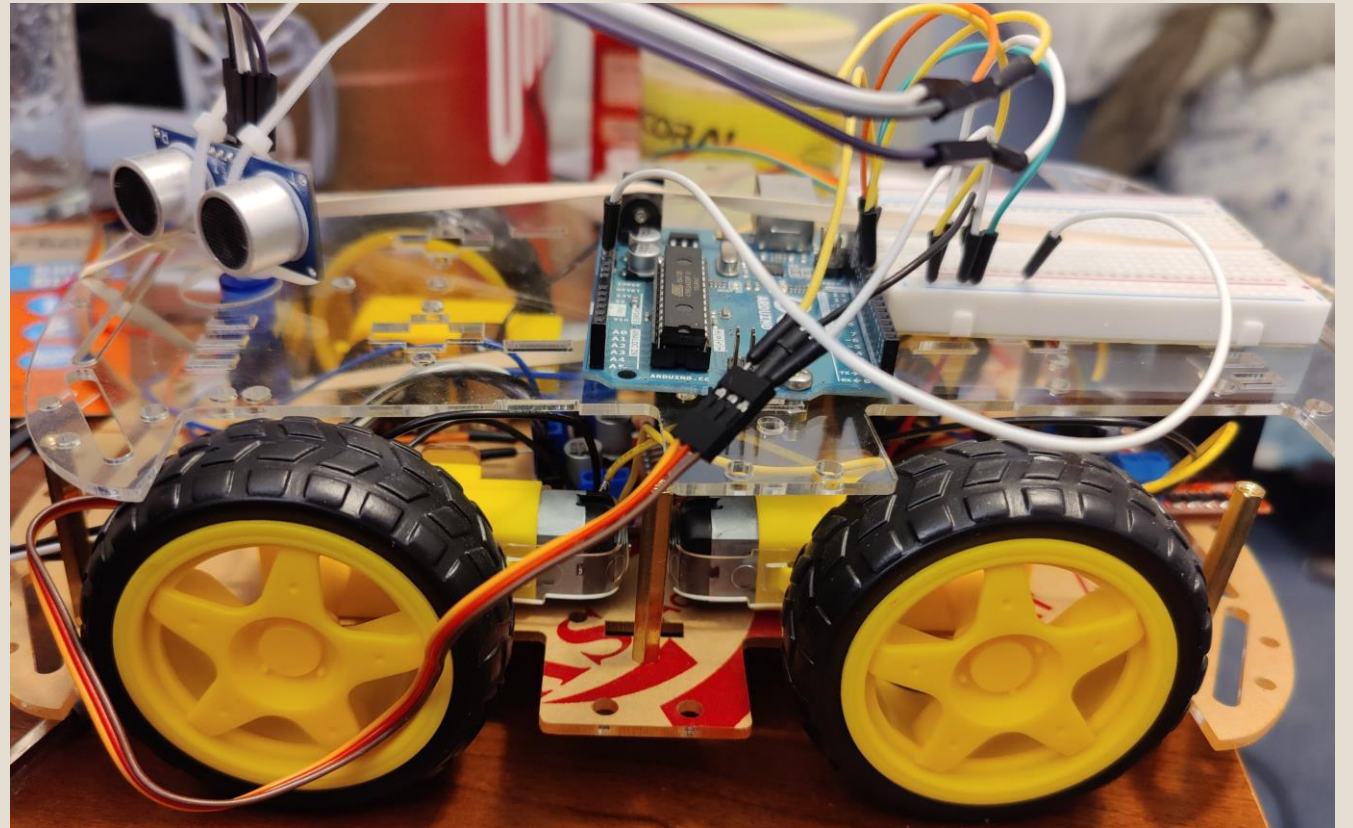


Programming Microcontrollers

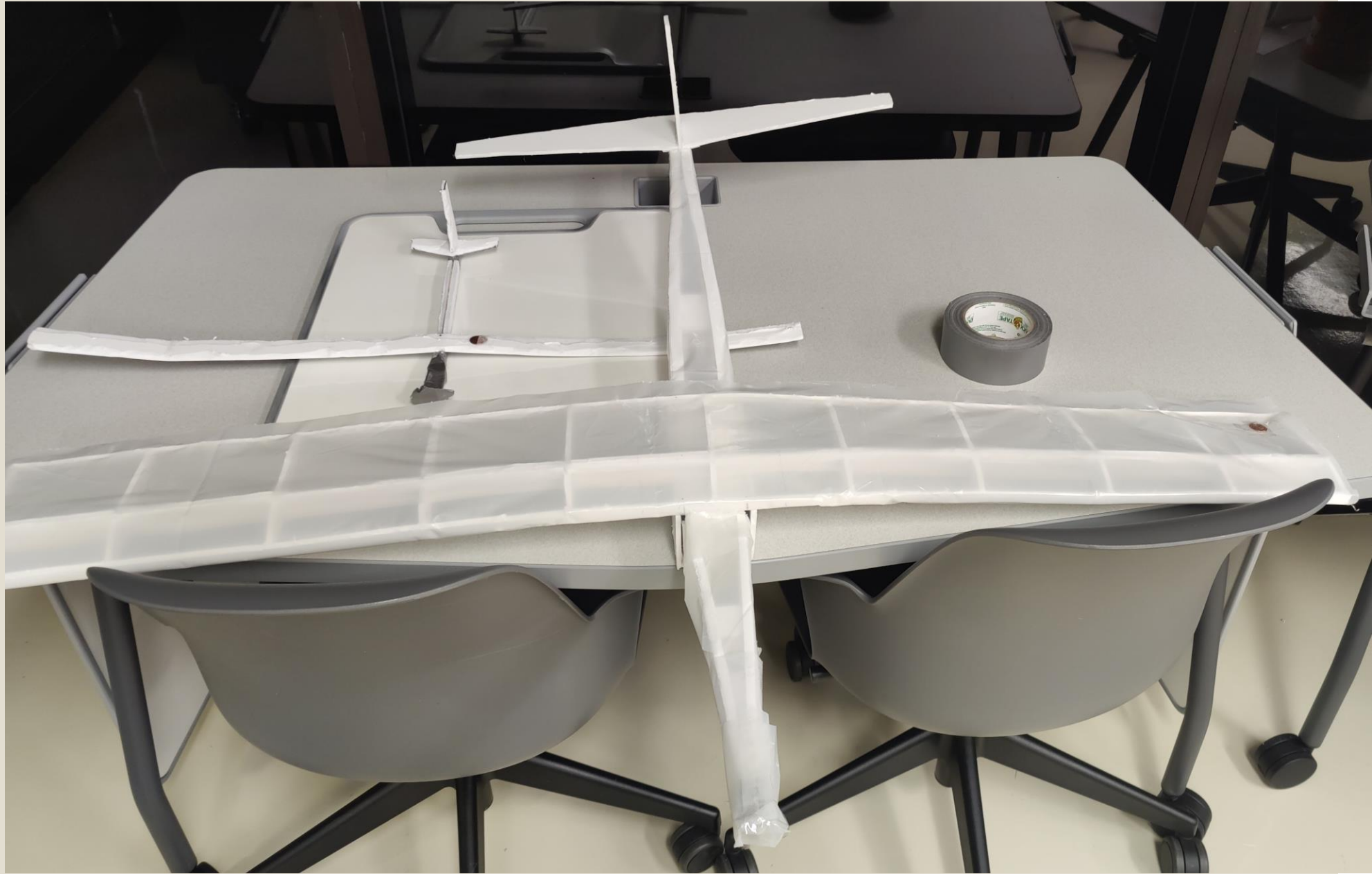
```
void servoSpin() { // Spins servo 60 degrees back and forth
  // i represents a degree the servo is positioned
  for (int i = center; i <= iLeftMax; i += i180spinspeed) {
    // left spin
    myServo.write(i); // turn servo
    pingStop(); // ping sensor
  }
  for (int i = iLeftMax; i >= iRightMax; i -= i180spinspeed) {
    // right spin
    myServo.write(i); // turn servo
    pingStop(); // ping sensor
  }
  for (int i = iRightMax; i <= center; i += i180spinspeed) {
    // back to middle
    myServo.write(i); // turn servo
    pingStop(); // ping sensor
  }
}

void pingStop() {
  int cmDist = threePing(); // figures out the distance in centimeter
  int forMapFunction = cmDist; // new value for map function
  if ( forMapFunction > mapFuncHighSpeed ) {
    // maximum value is 400. Car won't run if value goes above 400 with map function
    forMapFunction = mapFuncHighSpeed;
  }
  // 30cm = only 75 speed, 400cm = 200 speed
  rate = map(forMapFunction, mapFuncCmMin, mapFuncCmMax, mapFuncLowSpeed, mapFuncHighSpeed);

  if (cmDist > stopCmDist ) { // if distance greater than 40cm, go forward
    forward4(rate);
  } else { // below 40 cm
    straightBack(rateBack);
    off(); // turn off
  }
}
```



Glider



Assembling PCs



Code Example: C++, Python, Java

```
cout << "What is your dog's name? ";
cin >> nameOfDog;

cout << "What does the dog weigh in pounds? ";
cin >> sizeOfDog;

cout << "How many days per month will your dog need care? ";
cin >> daysPerMonth;

cout << "Pampared Pets Day Care Center\n\n";

// A control structure that uses selection to decide what the daily rate will be
// based on the weight of the dog
if (sizeOfDog < WEIGHT_MEDIUM_DOG)
{
    dailyRate += RATE_SMALL_DOG;
}
else if (sizeOfDog >= WEIGHT_BIG_DOG)
{
    dailyRate += RATE_BIG_DOG;
}
else
{
    dailyRate += RATE_MEDIUM_DOG;
}

// A discount is provided if days per month is over ten days. If a discount is provided,
// then the daily rate is reduced by 3 if the dog is a medium size, or 2 if any other size
if (daysPerMonth > DISCOUNT_FOR_OVER_TEN_DAYS)
{
    dailyRate -= (sizeOfDog >= WEIGHT_MEDIUM_DOG && sizeOfDog < WEIGHT_BIG_DOG) ?
        DISCOUNT_MEDIUM_DOG : DISCOUNT_OTHER_SIZE_DOG;
}

// Calculations to find the total bill
totalBill = dailyRate * daysPerMonth;

// Tells user the daily rate and total bill
cout << "The daily rate for " << nameOfDog << " is $" << dailyRate
    << "\nPer month, you will pay $" << totalBill;
```

```
class Vehicle: # class is created called vehicle

    # Initializer that sets speed to 0, and requires a vehicle's make and color
    def __init__(self, make, color):
        self.__speed = 0
        self.__make = make
        self.__color = color

    # creates hidden attributes which are the speed, make, and color

    def set_make(self, make): # Mutator Method that changes the make
        self.__make = make

    def get_make(self): # Accessor method that returns the make
        return self.__make

    def set_color(self, color): # Mutator Method that changes the color
        self.__color = color

    def get_color(self): # Accessor method that returns the color
        return self.__color

    def display_speed(self): # method that displays object's speed
        print('Current speed: ' + str(self.__speed))

    def accelerate(self): # method that lets the object's speed go up 1
        print('Accelerating...')
        self.__speed += 1

    def decelerate(self): # method that lets the object's speed go down 1
        print('Decelerating...')
        self.__speed -= 1

def main():

    # creates object, passes the make and color to the object
    vehicle = Vehicle(make='ford', color='black')

    for accel in range(2): # for loop that accelerates the vehicle object two times
        vehicle.accelerate()
        vehicle.display_speed()

    print()

    for deaccel in range(2): # for loop that deaccelerates the vehicle object two times
        vehicle.decelerate()
        vehicle.display_speed()

main()
```

```
public class Lab1
{
    // Initialized values to 0 for while loop.

    int bugs = 0, days = 0, percentage = 0;
    Scanner input = new Scanner(System.in);

    System.out.println("**** Bug Population Growth ****\n");

    // User must type in appropriate amount of bugs, percentage, and days.
    // If not, loop will print errors and ask user to retype values in.

    while (!(bugs >= 2 && percentage >= 0 && days >= 1))
    {
        System.out.print("How many bugs are present on day 0? ");
        bugs = input.nextInt();

        System.out.print("What percentage of daily bug growth is expected? ");
        percentage = input.nextInt();

        System.out.print("How many days will the simulation run? ");
        days = input.nextInt();

        // Determine which error messages print if user did not type in appropriate numbers
        if (!(bugs >= 2 && percentage >= 0 && days >= 1))
        {
            System.out.println("\nInput error:");

            if (!(bugs >= 2))
            {
                System.out.println("There must be 2 or more bugs on day 0");
            }
            if (!(percentage >= 0))
            {
                System.out.println("The percentage of daily growth must be positive.");
            }
            if (!(days >= 1))
            {
                System.out.println("The simulation must run for at least 1 day.");
            }

            System.out.println("Try again!\n");
        }
    }

    System.out.print("\n");

    // for loop calculate how many bugs are in the colony each day and prints that
    // number for each day. Calculations truncate decimal values since you cannot
    // have a fraction of a bug.

    for (int i = 0; i < days; ++i)
    {
        bugs += (int) (bugs * percentage * 0.01);

        System.out.printf("On day %d there will be %d bugs.\n", i+1, bugs);
    }
    input.close();
}
```

High School Band Awards

