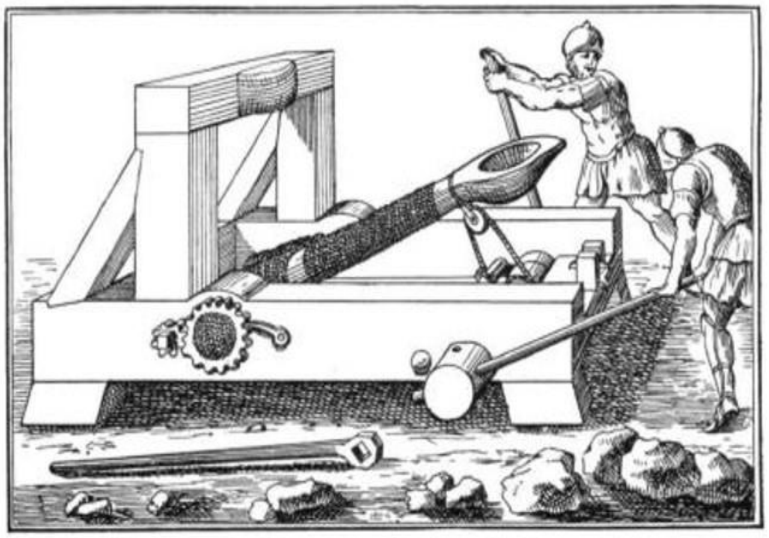
DEPARTMENT

OF

ELECTRONICS AND COMMUNICATION ENGINEERING

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

# Handout/Assignment-
for
Engineering Design Project-I (UTA013)



INSTRUCTOR INCHARGE

Name – Meharamt Singh
Roll No. - 102003241

ASSIGNMENT – 3 (ASSEMBLY OF HARDWARE AND SOFTWARE)

Exercise 1 – To construct a circuit which will record and display the length of time taken by the throwing arm on the Mangonel to pass between the sensors mounted on the chassis. Implement the real hardware on the breadboard, at the input side of the Arduino. Demonstrate the hardware and software by attaching to a real Mangonel.

Hardware Required

- CD4027, CD4081, CD4543
- Arduino Uno
- Seven Segment Display
- Single core connecting wires
- Digital Trainer Kit
- Tinkercad Software tool (https://www.tinkercad.com/)

Theory

The circuit is required to:

- Accept the individual inputs from the two sensors on the Mangonel,
- Combine the two signals into one,
- Convert the two short pulses into one long pulse,
- Deliver this pulse to the relevant input on the Arduino Board,

The input signals:

The input signals received from the sensors on the Mangonel are shown in Figure 1 below:

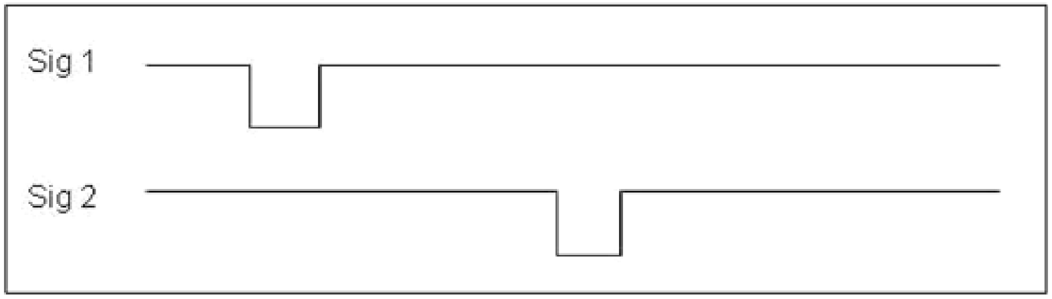


Figure 1: Mangonel sensors signals

Required signal:

The signal required for the Arduino program is shown in figure 2:

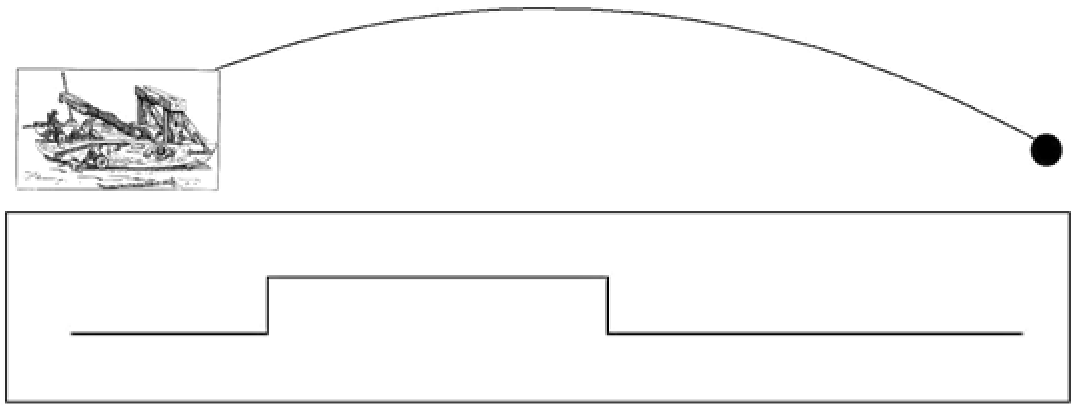


Figure 2: Required signal for Arduino.

How do we achieve the required signal?

In order to make one signal out of the two, we first need to combine the two individual signals. This is done by means of a combination logic setup. You should by now be familiar with the basic logic circuits as shown below.

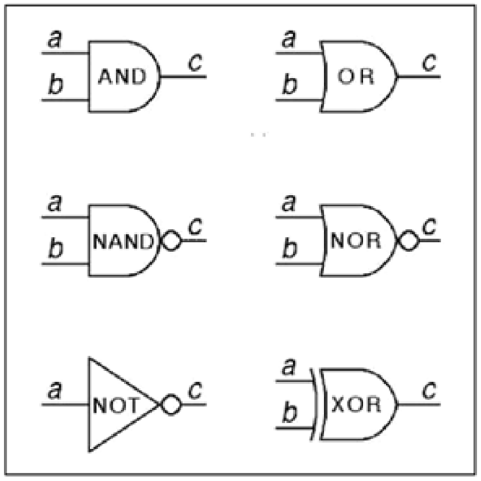


Figure 3: Basic logic gates

Of the devices above, the best choices for combining our two signals into one are the AND or NAND gates as either will preserve the signal from each individual sensor whilst also combining the two signals into one. It is important to note that NAND is simply inverted AND, similarly, if we invert NAND, we get AND again.

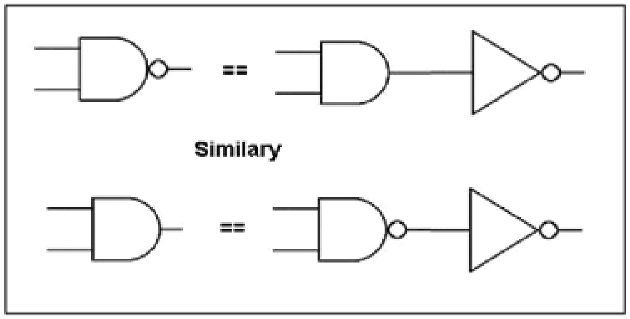


Figure 4: Logic inversion

The output from NAND and AND gates respectively to the input signal provided are shown below.

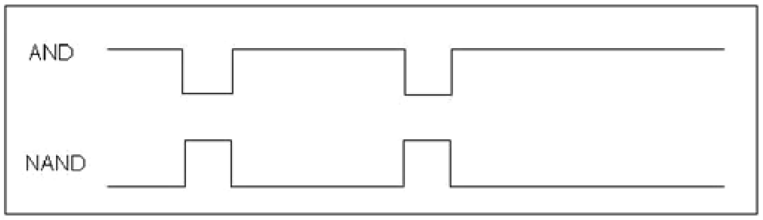


Figure 5: AND and NAND logic outputs

Figure 5 shows how the logic signals from the inputs are each maintained but combined into one signal. The next task is to convert the two small pulses into one long pulse.

Looking at the AND output signal, it could be argued that we already have a long pulse from this step alone – however, the long pulse between the two short pulses given by the AND logic is not an accurate measurement of the transit time of the arm because the pulse length is shortened by the width of the second pulse. We need to generate a single long pulse as shown in figure 6 which extends from either the rising or falling edge of the first pulse, to the corresponding edge of the second pulse.

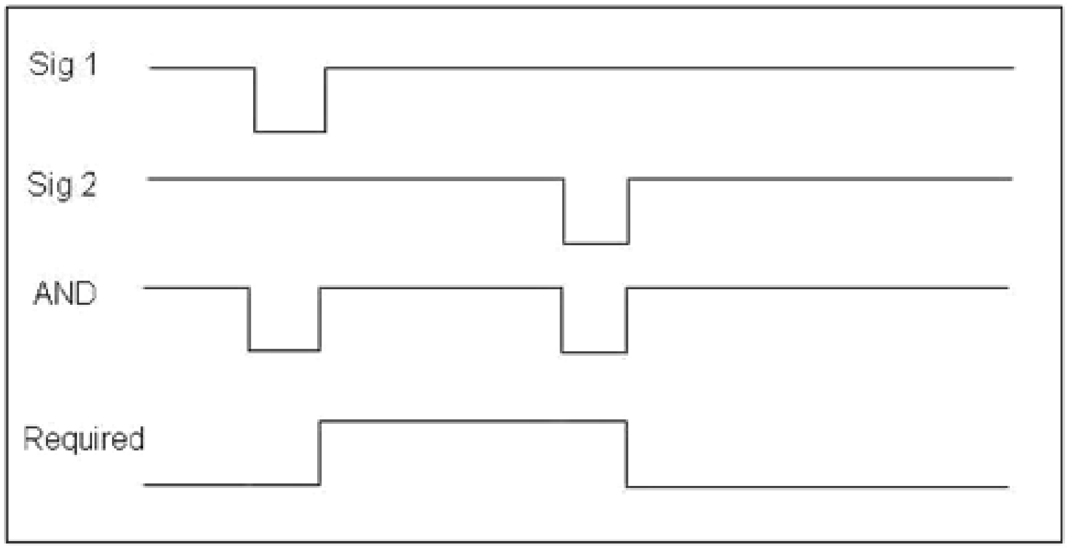
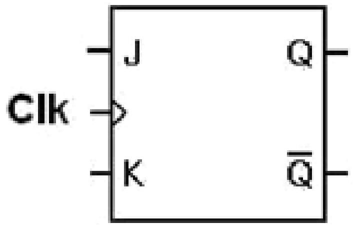
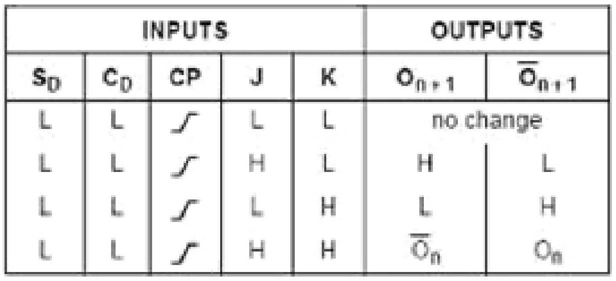


Figure 6: Logic comparison chart

This can be achieved by using the two shorter pulses in the output of the AND gate to “toggle” or “clock” the output of a latching logic device. For this purpose, we are going to use a J-K flip-flop.





INPUTS					OUTPUTS	
S_D	C_D	CP	J	K	Q_{n+1}	$\overline{Q}_{n+1}$
L	L	↑	L	L	no change	
L	L	↑	H	L	H	L
L	L	↑	L	H	L	H
L	L	↑	H	H	$\overline{Q}_n$	Q_n

Figure 7: J-K flipflop and function table.

The J -K flip- flop is the most versatile of the basic flip -flops. It has two inputs, traditionally labelled J and K. If J and K are different then the output Q takes the value of J at the next clock edge. If J and K are both low then no change occurs. If J and K both are high at the clock edge then the output will toggle from one state to the other.

This toggle application means that we can use the two signal pulses from the output of the AND gate to cause the output of the flip-flop to change state, hence allowing us to create one long pulse between the corresponding edges of the two shorter pulses. The device also has set and reset functions which will be useful for clearing the output states when we want to start a new timing run. Figure 7 shows the function table of the J-K flip-flop.

The output of out J-K flip-flop then goes to the appropriate input of the Arduino and from there the program in the Arduino will measure the length of the pulse and output the appropriate values to the various registers on the display. Hence, our circuit should wind up looking something like this:

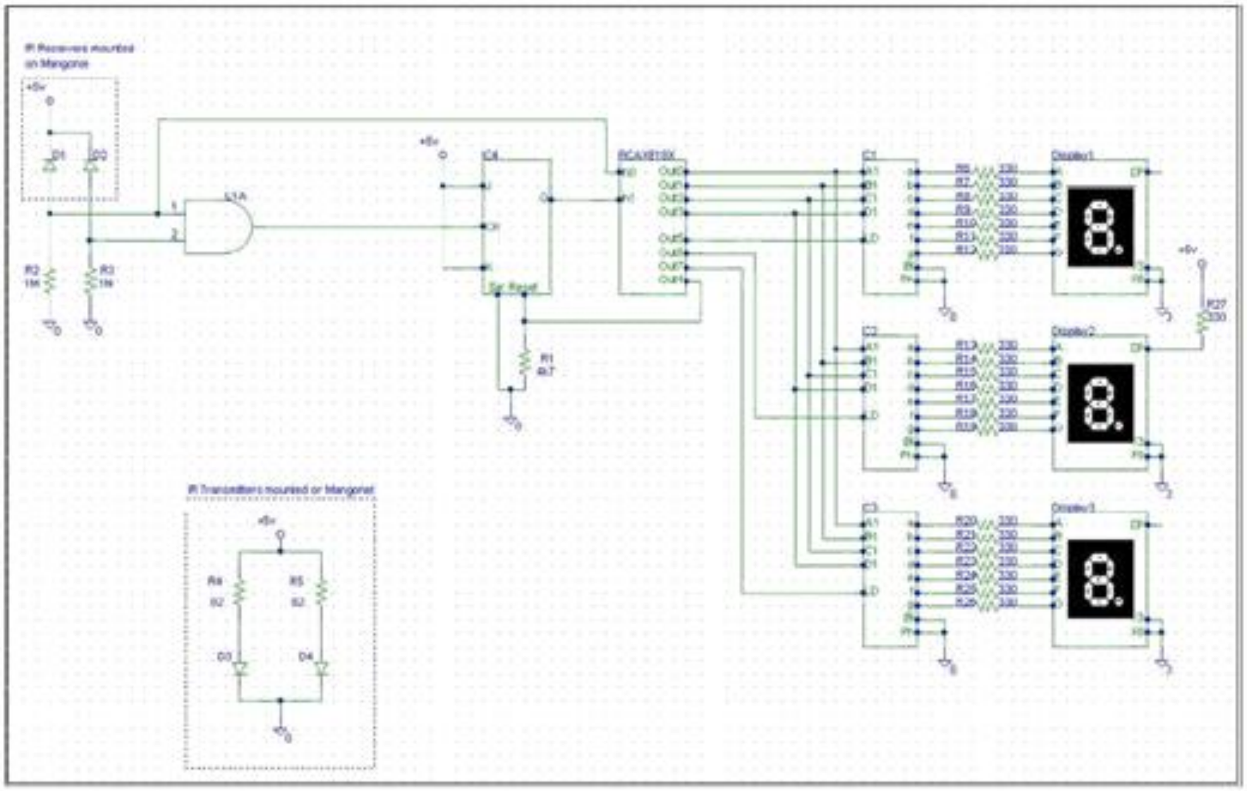
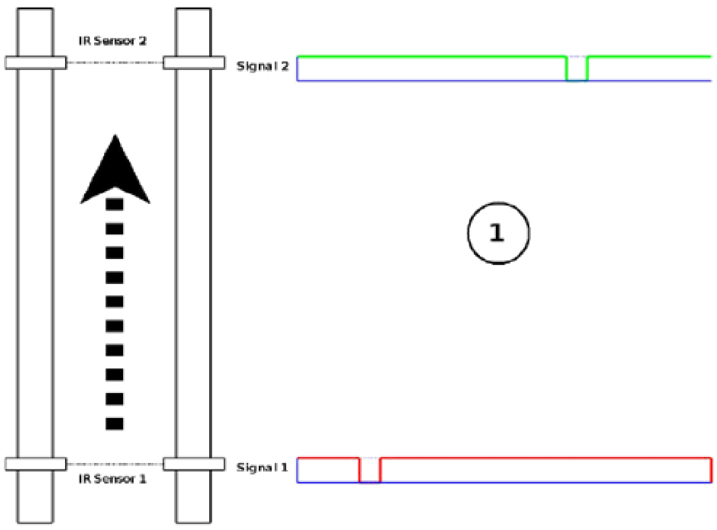

IR Receivers mounted on Mangonel
IR Transmitters mounted on Mangonel
Display1
Display2
Display3

Figure 8: Circuit Schematic

Explain Figures 9 succinctly, in fewer than 50 words per stage, each of the five stages involved in measuring the swing time of the Mangonel arm.



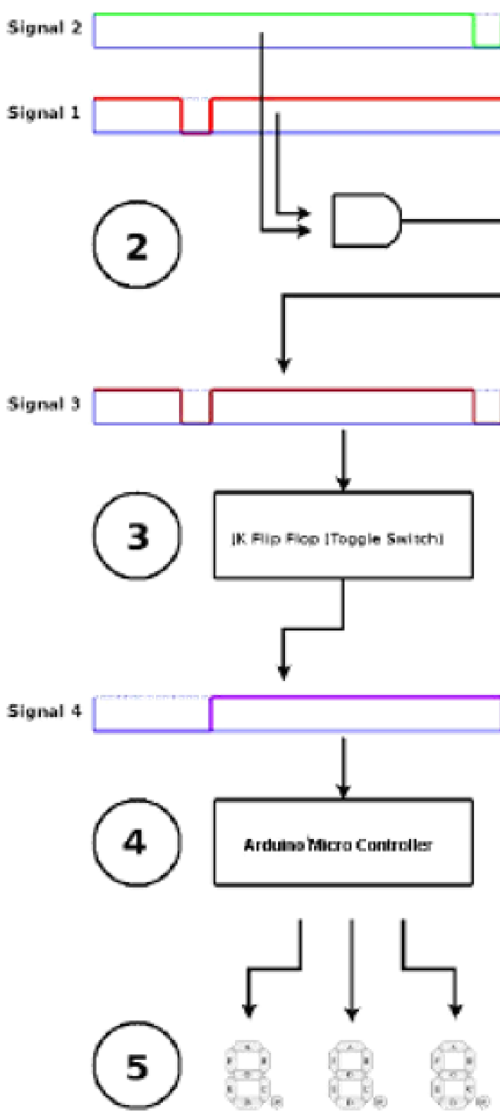
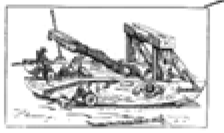
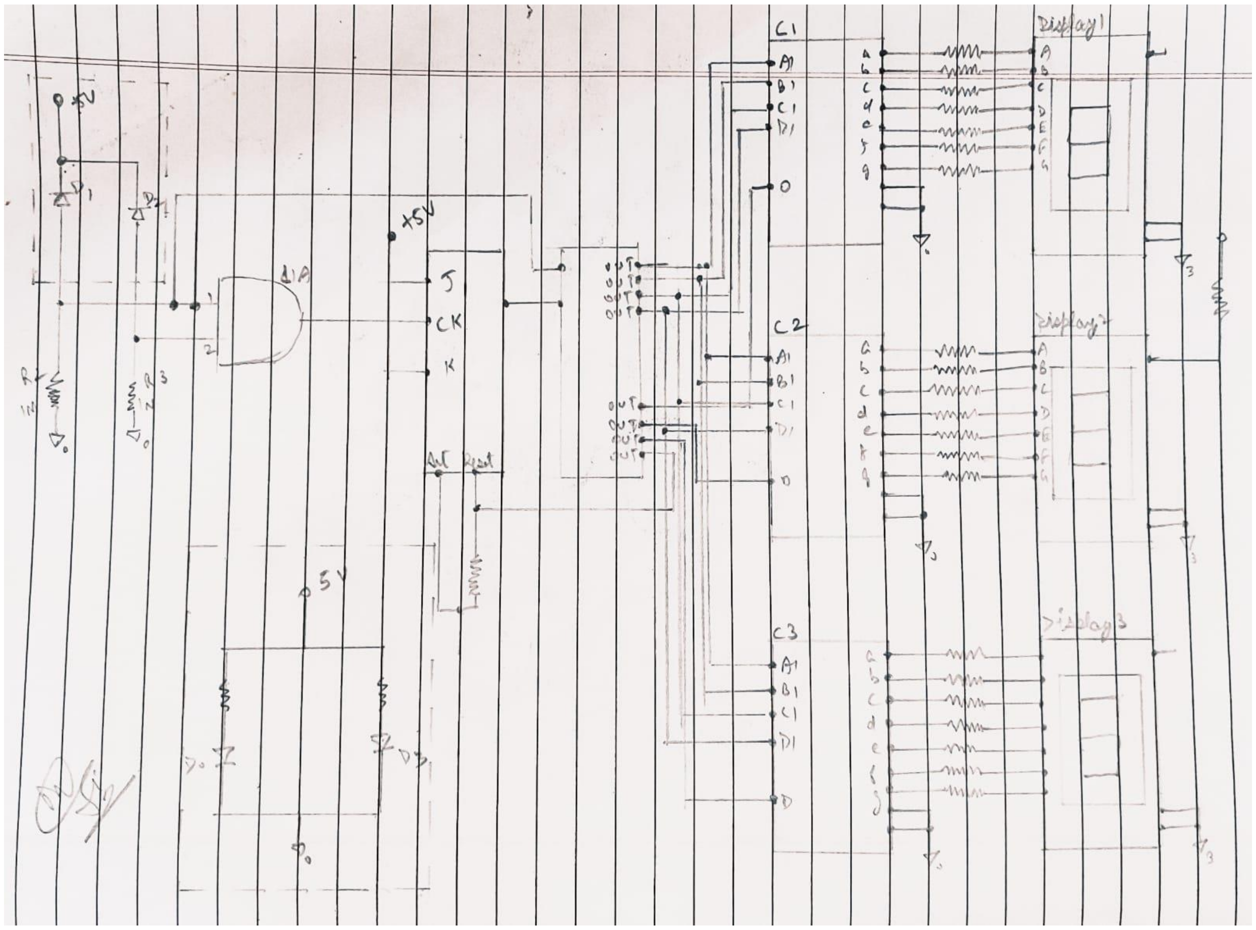


Figure 9: Stages 1 to 5



Schematic Diagram:

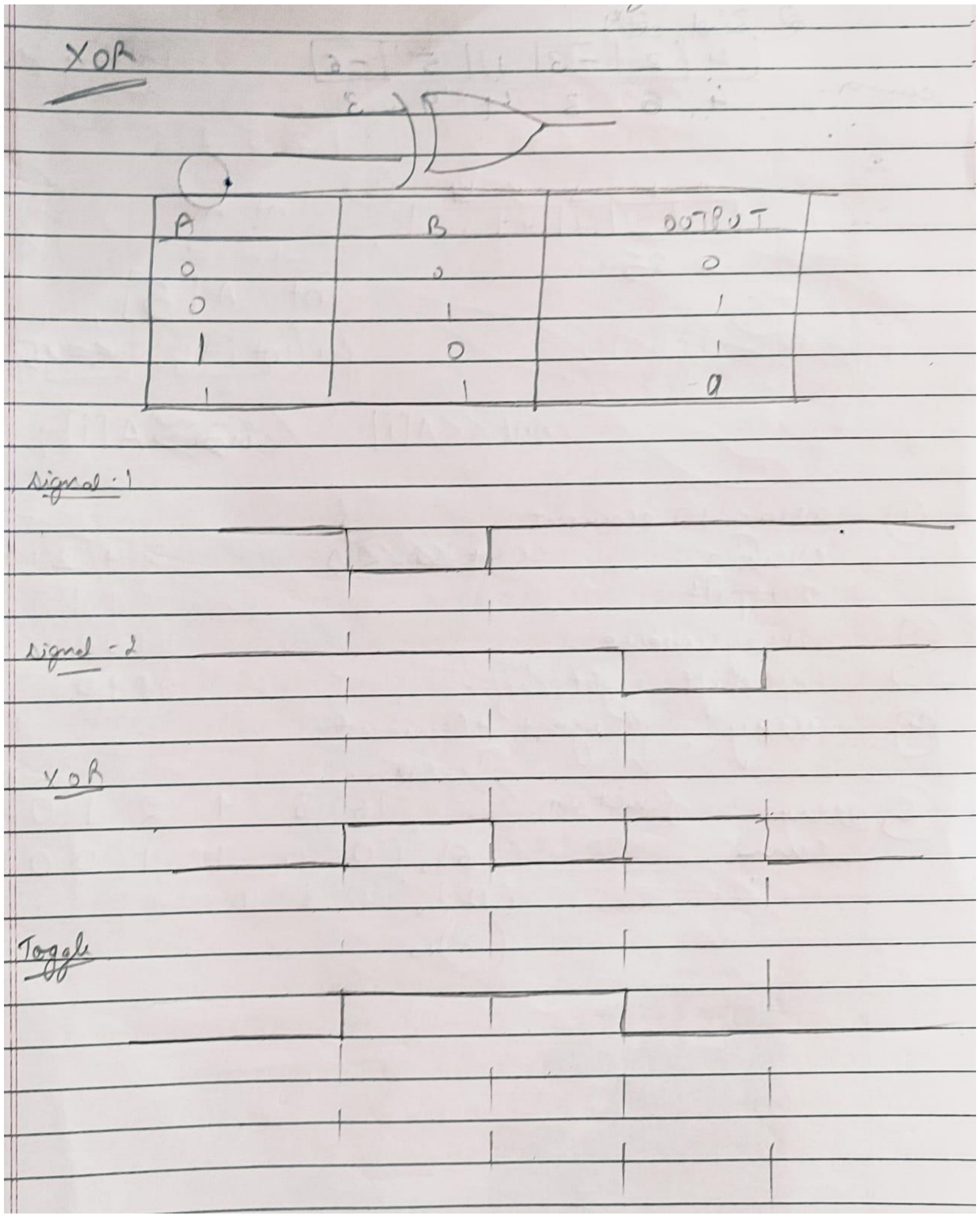


Reflection:

- When the moving arm crosses an IR sensor, a HIGH pulse is generated and both the signals from the 2 IR sensors are fed into the input of AND gate.
- The output of the AND gate acts as a CLK signal for the JK Flip Flop. The output of JK Flip Flop gives the HIGH pulse for a time equal to the time taken by the moving arm.
- The output of JK flip flop goes to the appropriate input of the Arduino and from there the program in the Arduino will measure the length of the pulse and output the appropriate values on the 7-segment Displays.

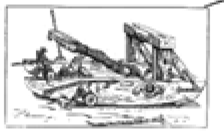
Assignment Tasks:

1. Obtain the required signal for Arduino shown in figure 2 using at least two different logic gates (explain using waveforms).



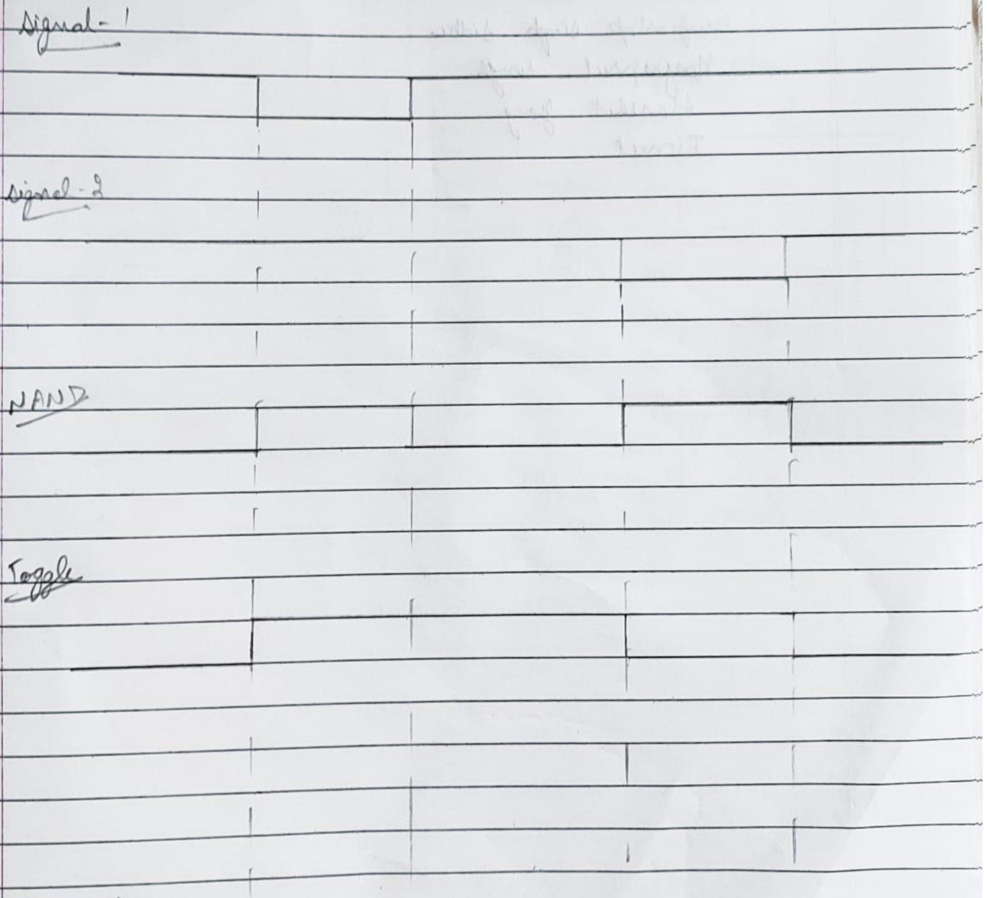
XOR

A	B	OUTPUT
0	0	0
0	1	1
1	0	1
1	1	0



NAND

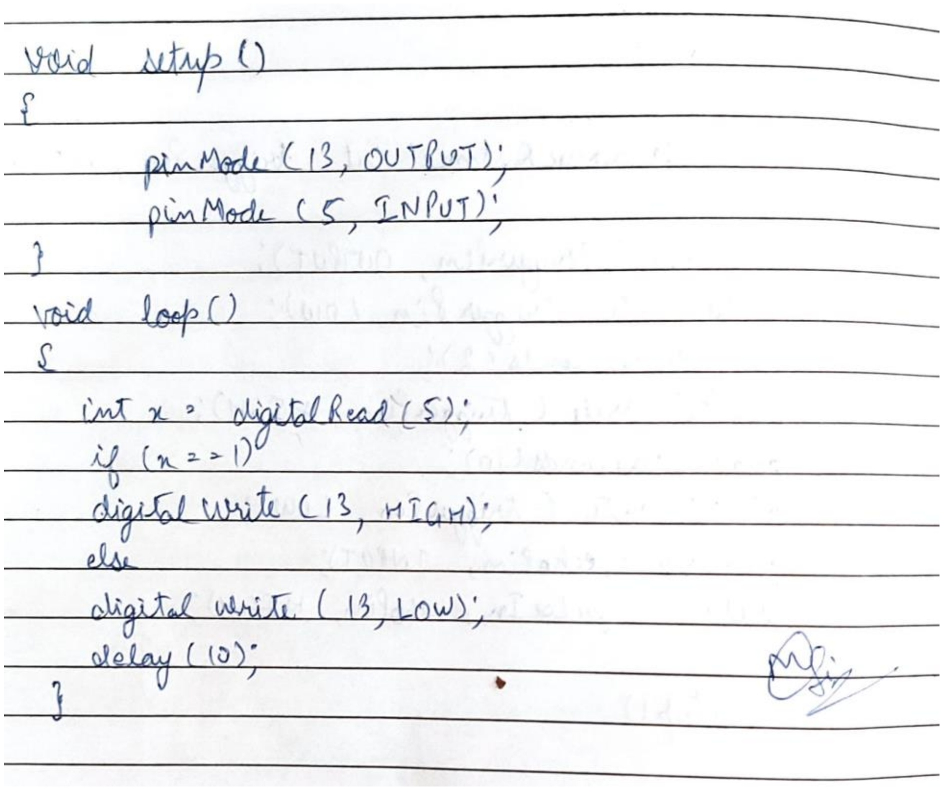
A	B	OUTPUT
0	0	1
0	1	1
1	0	1
1	1	0



[illegible]

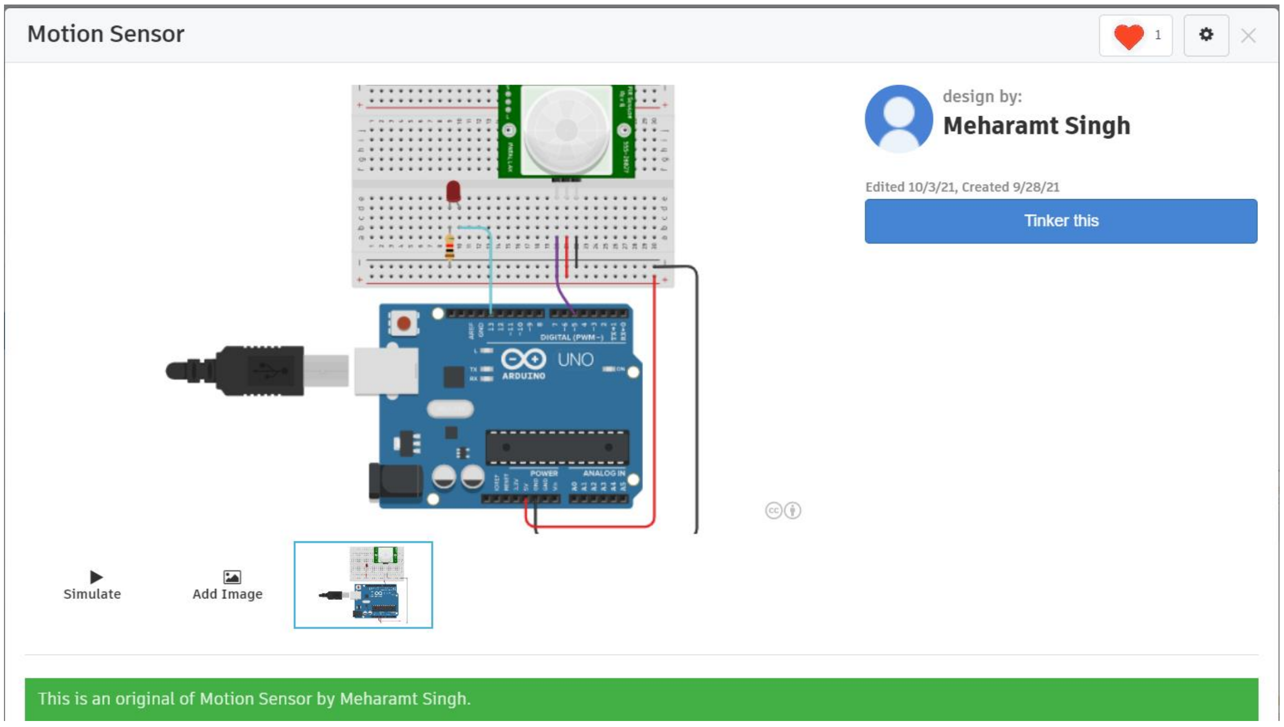
2. Using Tinkercad, design the following sensor based micro-projects to:

    a. Detect the motion of an object, and



```
void setup()
{
        pinMode(13, OUTPUT);
        pinMode(5, INPUT);
}
void loop()
{
    int x = digitalRead(5);
    if (x==1)
    digital write(13, HIGH);
    else
    digital write(13, LOW);
    delay(10);
}
```

[illegible]

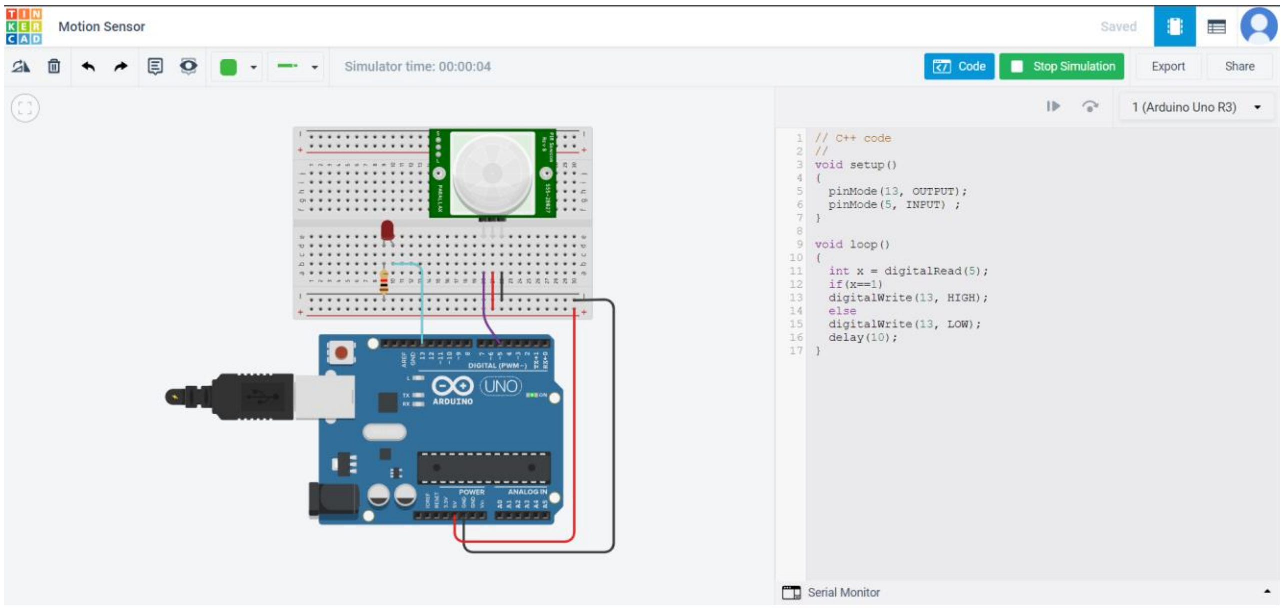


Motion Sensor

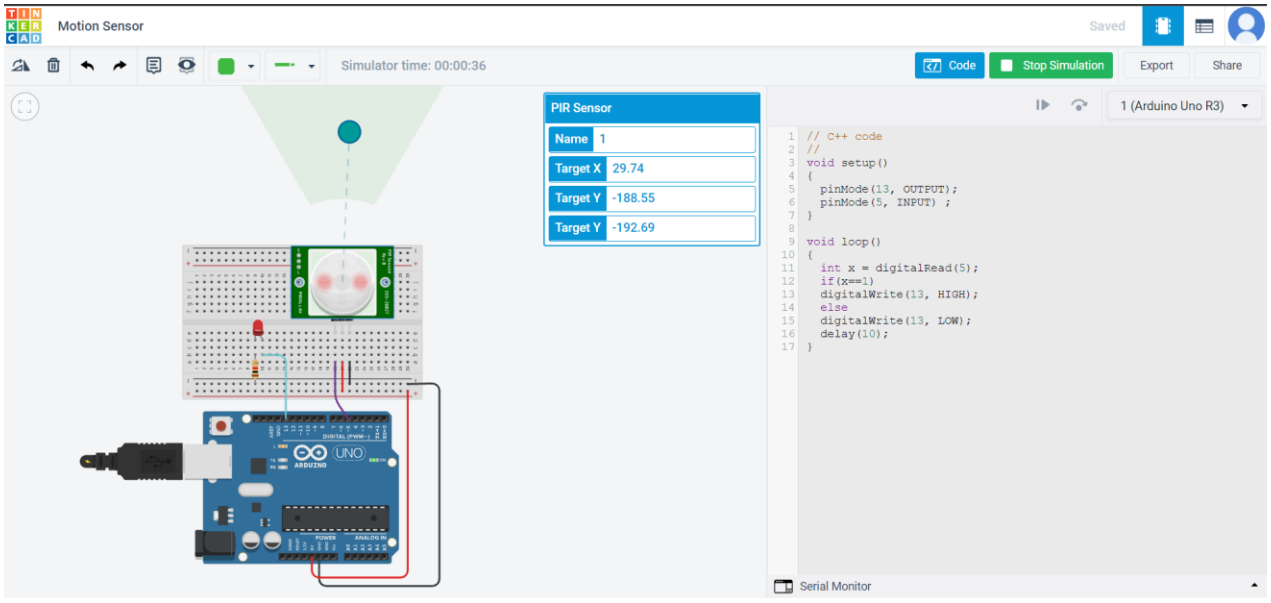
design by:
Meharamt Singh

Edited 10/3/21, Created 9/28/21

This is an original of Motion Sensor by Meharamt Singh.



```
// C++ code
//
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(5, INPUT) ;
}

void loop()
{
  int x = digitalRead(5);
  if(x==1)
  digitalWrite(13, HIGH);
  else
  digitalWrite(13, LOW);
  delay(10);
}
```



PIR Sensor

Name	1
Target X	29.74
Target Y	-188.55
Target Y	-192.69

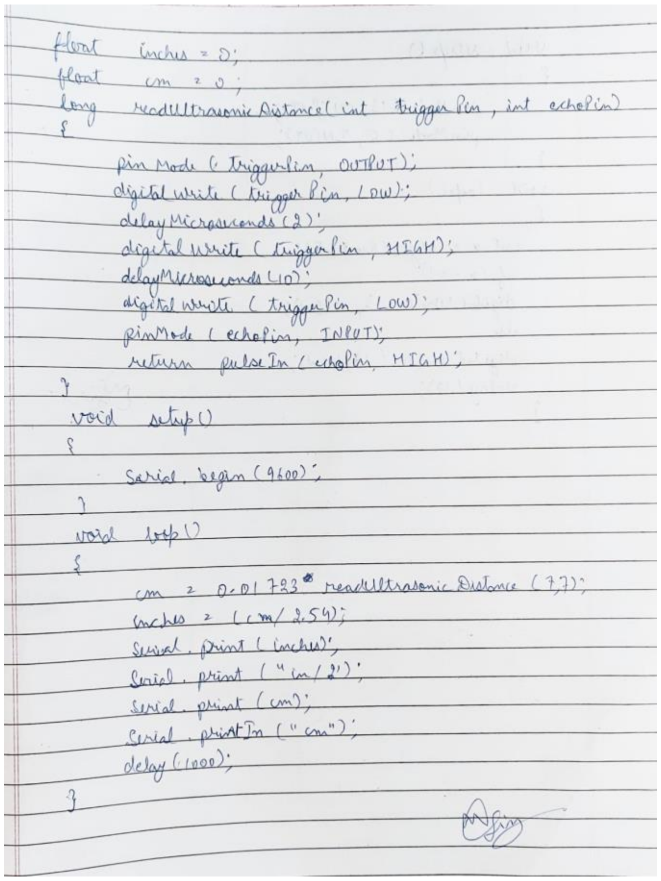
```
// C++ code
//
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(5, INPUT) ;
}

void loop()
{
  int x = digitalRead(5);
  if(x==1)
  digitalWrite(13, HIGH);
  else
  digitalWrite(13, LOW);
  delay(10);
}
```

Applications:

● The PIR/IR Sensor which is used to detect the motion of an object can be used in the mangonel to detect the motion of the throwing arm.

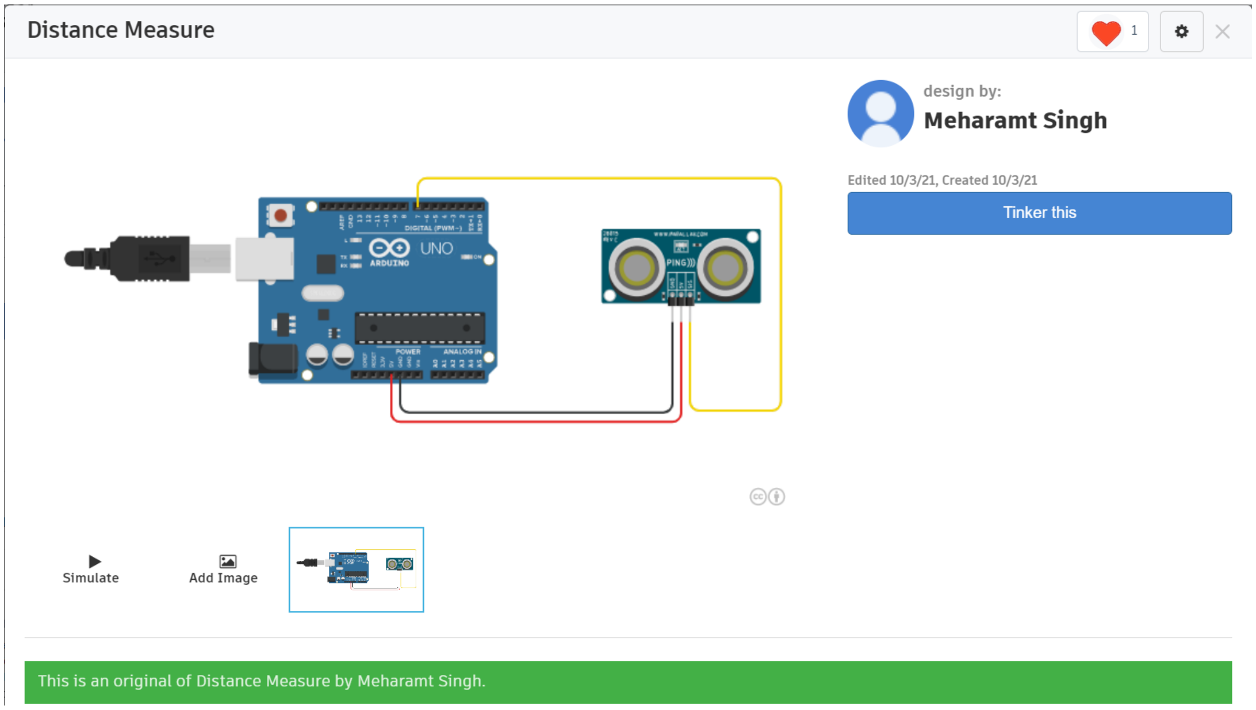
● It can be used to increase the precision of the mangonel so as to obtain maximum velocity and range from the throwing arm.

b. Measure distance between an object and the sensor itself.



```
float   inches = 0;
float   cm = 0;
long   readUltrasonicDistance( int  triggerPin , int  echoPin)
{
      pin Mode ( TriggerPin,  OUTPUT);
      digital write ( trigger Pin, LOW);
      delayMicroseconds (2);
      digital write ( triggerPin , HIGH);
      delayMicroseconds (10);
      digital write  ( triggerPin,  LOW);
      pinMode ( echoPin,  INPUT);
      return  pulseIn ( echoPin, HIGH);
}
 void  setup()
 {
      Serial. begin (9600);
  }
 void  loop()
 {
      cm  =  0.01723* readUltrasonicDistance (7,7);
      inches  =  (cm/ 2.54);
      Serial. print ( inches);
      Serial. print ( "in/ 2");
      Serial. print (cm);
      Serial. printIn ("cm");
      delay (1000);
 }
```

[illegible]



Distance Measure

1

design by:

Meharamt Singh

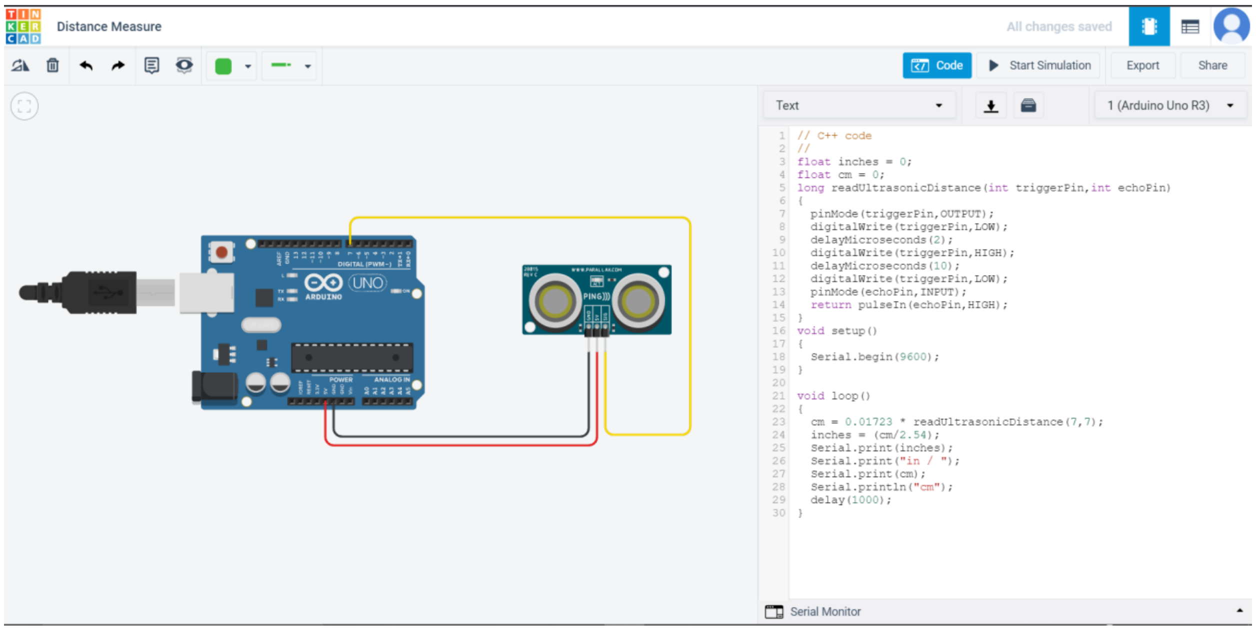
Edited 10/3/21, Created 10/3/21

Tinker this

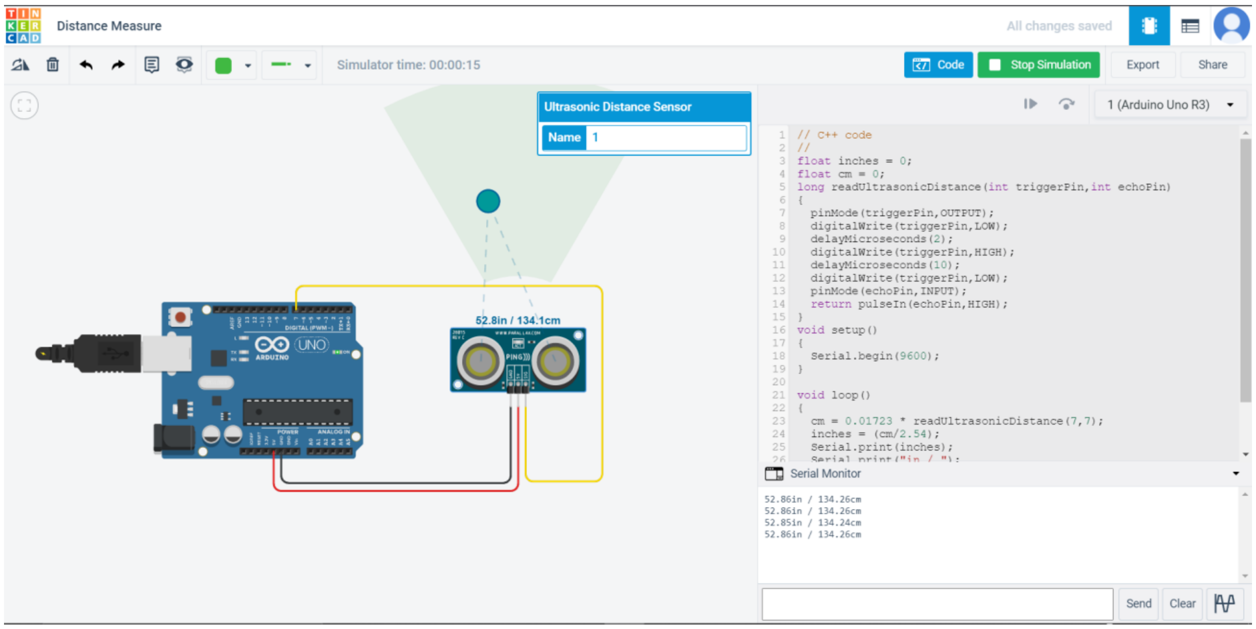
Simulate

Add Image

This is an original of Distance Measure by Meharamt Singh.



```
// C++ code
//
float inches = 0;
float cm = 0;
long readUltrasonicDistance(int triggerPin,int echoPin)
{
  pinMode(triggerPin,OUTPUT);
  digitalWrite(triggerPin,LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin,LOW);
  pinMode(echoPin,INPUT);
  return pulseIn(echoPin,HIGH);
}
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  cm = 0.01723 * readUltrasonicDistance(7,7);
  inches = (cm/2.54);
  Serial.print(inches);
  Serial.print("in / ");
  Serial.print(cm);
  Serial.println("cm");
  delay(1000);
}
```



```
// C++ code
//
float inches = 0;
float cm = 0;
long readUltrasonicDistance(int triggerPin,int echoPin)
{
  pinMode(triggerPin,OUTPUT);
  digitalWrite(triggerPin,LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin,LOW);
  pinMode(echoPin,INPUT);
  return pulseIn(echoPin,HIGH);
}
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  cm = 0.01723 * readUltrasonicDistance(7,7);
  inches = (cm/2.54);
  Serial.print(inches);
  Serial.print("in / ");
```

```
52.86in / 134.26cm
52.86in / 134.26cm
52.85in / 134.24cm
52.86in / 134.26cm
```

Applications:

● The ultrasonic sensor is used to measure the distance between the sensor and the object can be used in the mangonel to measure the distance between the sensor and the throwing arm.

● It would give accurate results and would help in increasing the precision of the mangonel so as to obtain the maximum velocity and range covered by the object.