

ARIES

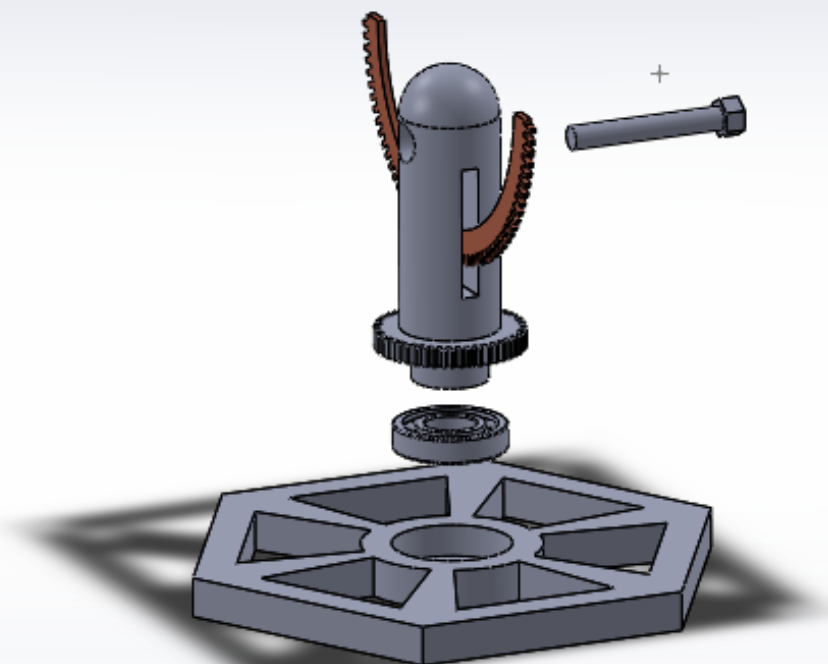
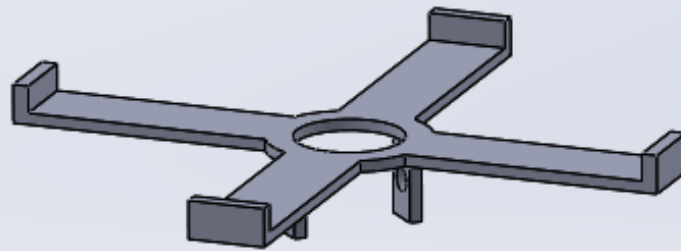
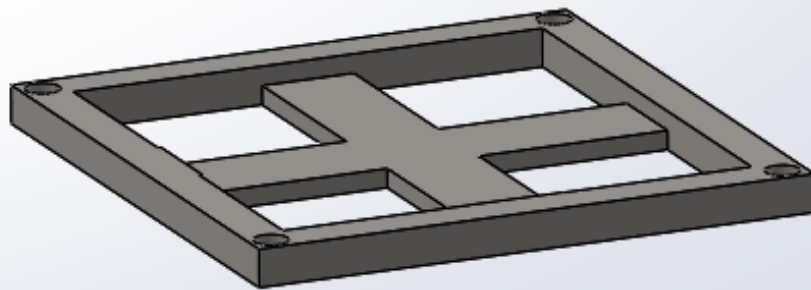
AUDRINO SOLAR PANEL

PROBLEM STATEMENT

A solar panel which can rotate in the direction of Sun in 2 doFs.

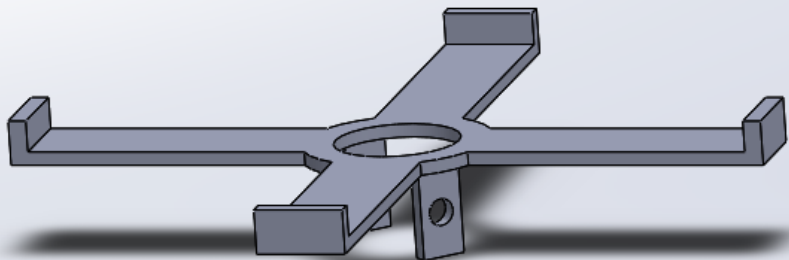
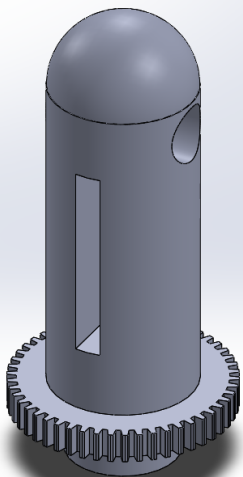
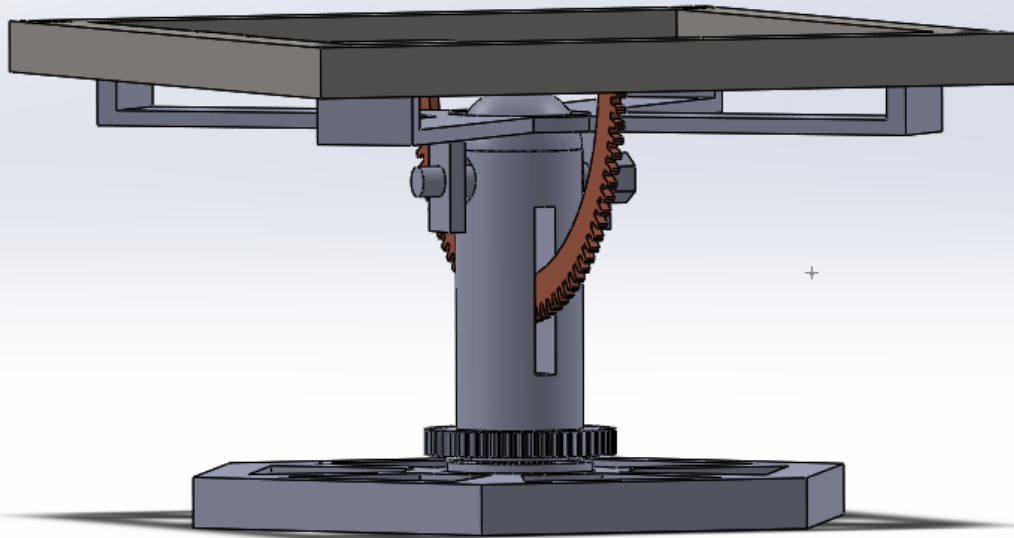
Project

Mechanical Design



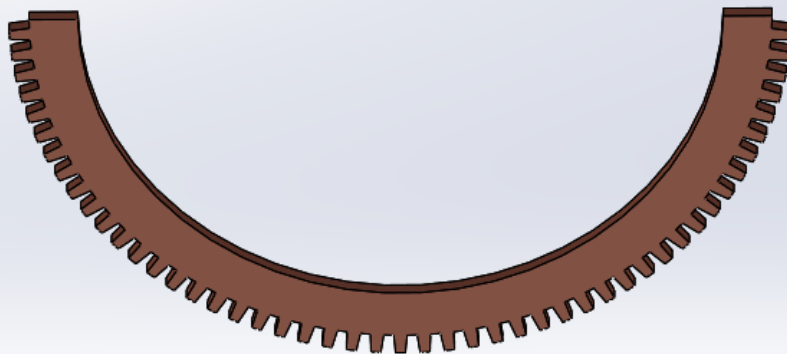
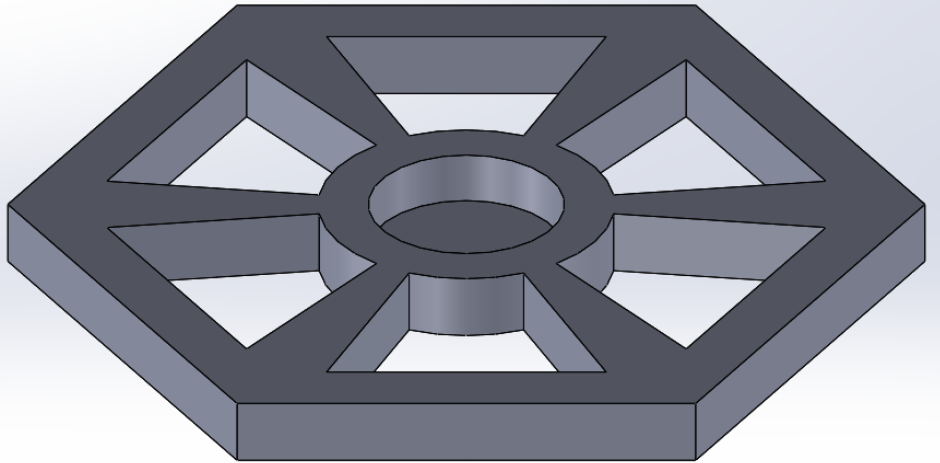
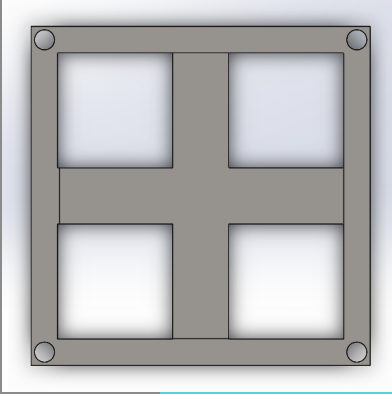
Project

Mechanical Design



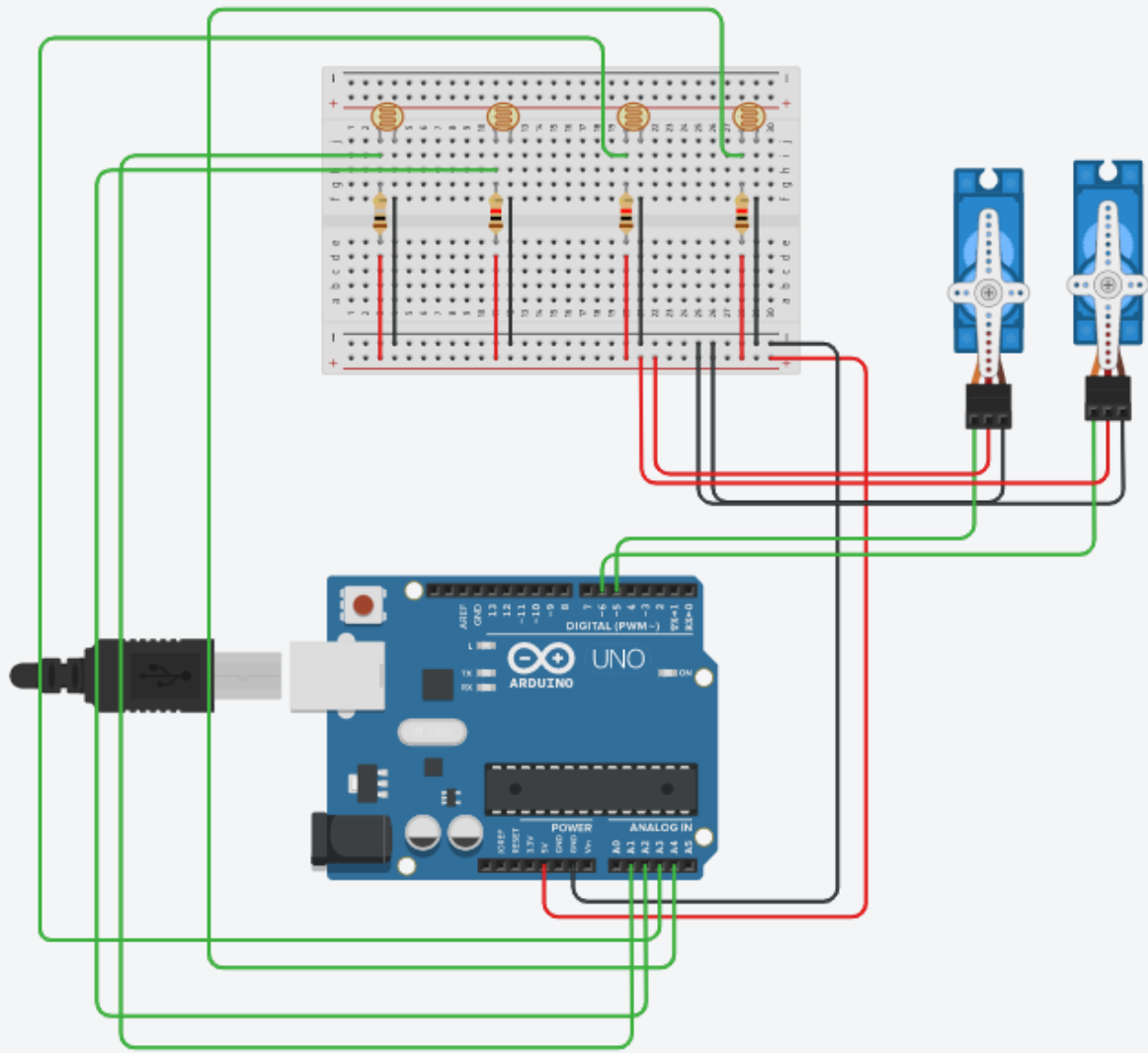
Project

Mechanical Design



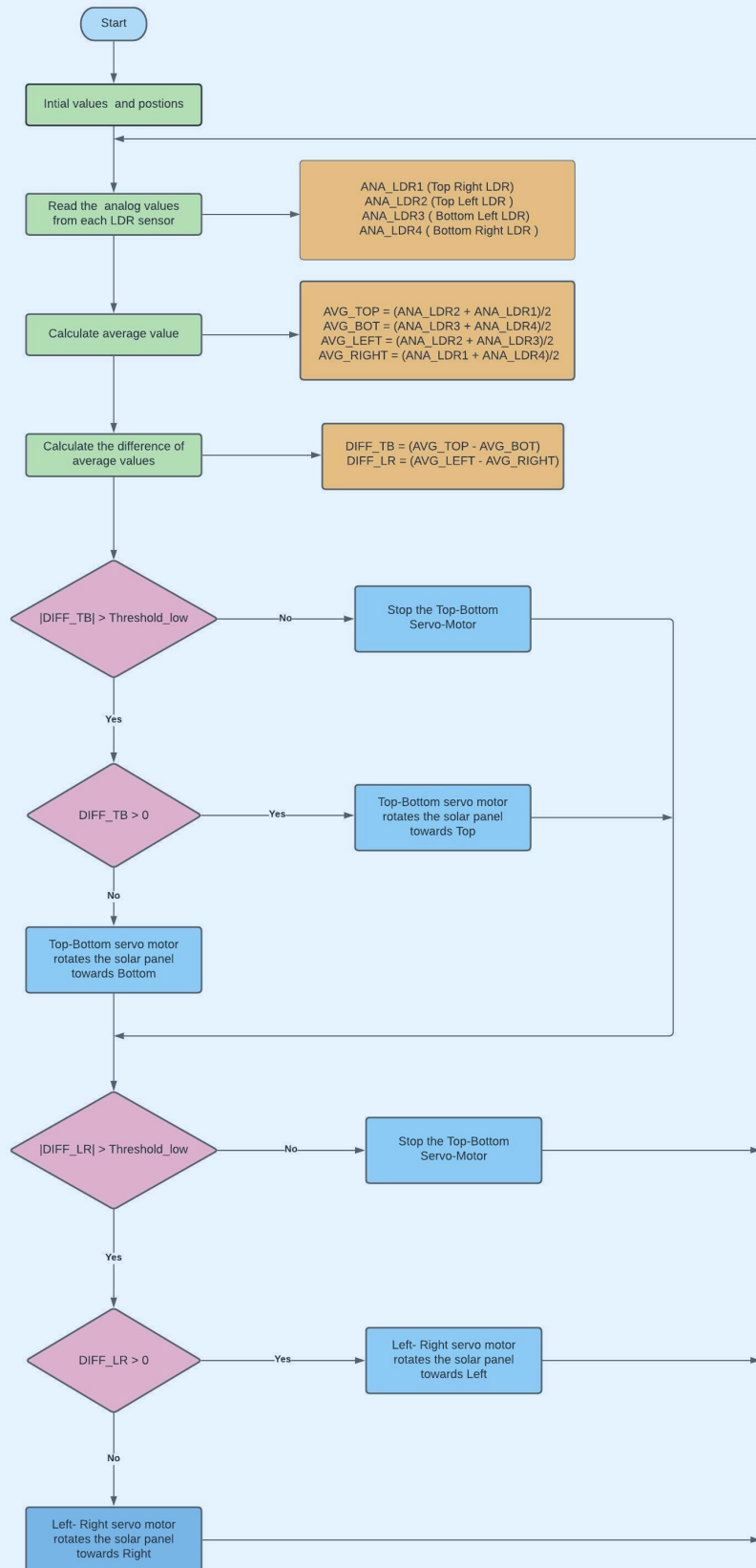
Project

Electronics Design



Project

Algorithm



Project

Programing

```
#include <Servo.h>
```

```
Servo SMTB;  
int SMTB_Angle = 0;
```

```
Servo SMLR;  
int SMLR_Angle = 0;
```

```
int LDR2 = 2; //top left LDR  
int LDR1 = 1; //top right LDR  
int LDR3 = 3; // bottom left LDR  
int LDR4 = 4; // bottom right LDR
```

```
int Threshold_low = 20;
```

```
void setup () {  
  SMTB.attach(5);  
  SMTB.write(0);
```

Project

Programing

```
SMLR.attach(6);  
SMLR.write(0);  
    delay(500);  
}
```

```
void loop() {  
    SMTB_Angle = SMTB.read();  
    SMLR_Angle = SMLR.read();
```

```
    int ANA_LDR1 = analogRead(1);  
    int ANA_LDR2 = analogRead(2);  
    int ANA_LDR3 = analogRead(3);  
    int ANA_LDR4 = analogRead(4);
```

```
    int AVG_TOP = (ANA_LDR2 + ANA_LDR1)/2;  
    int AVG_BOT = (ANA_LDR3 + ANA_LDR4)/2;  
    int AVG_LEFT = (ANA_LDR2 + ANA_LDR3)/2;  
    int AVG_RIGHT = (ANA_LDR1 + ANA_LDR4)/2;
```

```
    int DIFF_TB = (AVG_TOP - AVG_BOT);  
    int DIFF_LR = (AVG_LEFT - AVG_RIGHT);
```


Project

Programing

```
if(abs(DIFF_TB)> Threshold_low){  
    if(DIFF_TB > 0 ){  
        SMTB.write(SMTB_Angle + 1);  
        delay(100);  
    }  
    else {  
        SMTB.write(SMTB_Angle - 1);  
    }  
    } else {  
        SMTB.write(SMTB_Angle);  
    }  
if(abs(DIFF_LR)> Threshold_low){  
    if(DIFF_LR > 0 ){  
        SMLR.write(SMLR_Angle + 1);  
        delay(100);  
    }  
    else {  
        SMLR.write(SMLR_Angle - 1) ;  
    }  
    }else {  
        SMLR.write(SMLR_Angle);  
    }  
}
```