

Automated Light Focusing

Lachlan Hooper

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

Abstract

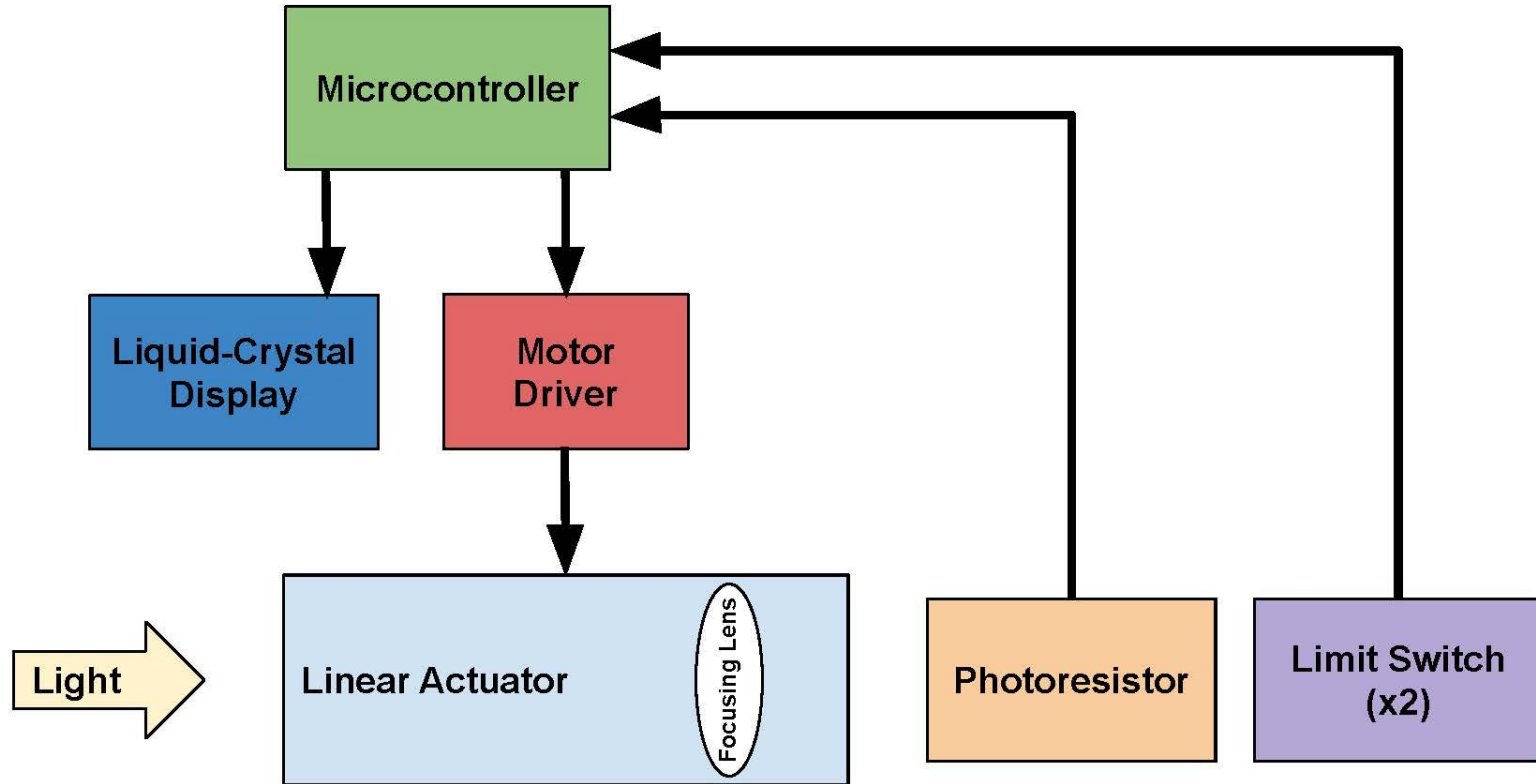
- Moving a convex focusing lens relative to a light source can result in light with a greater intensity
- But such manipulation can be tedious
- Lens moves to one end of range
- Scans all possible values and records them
- Maximum value is calculated
- Linear actuator moves to maximum intensity position
- Information about the calibration is displayed

Parts

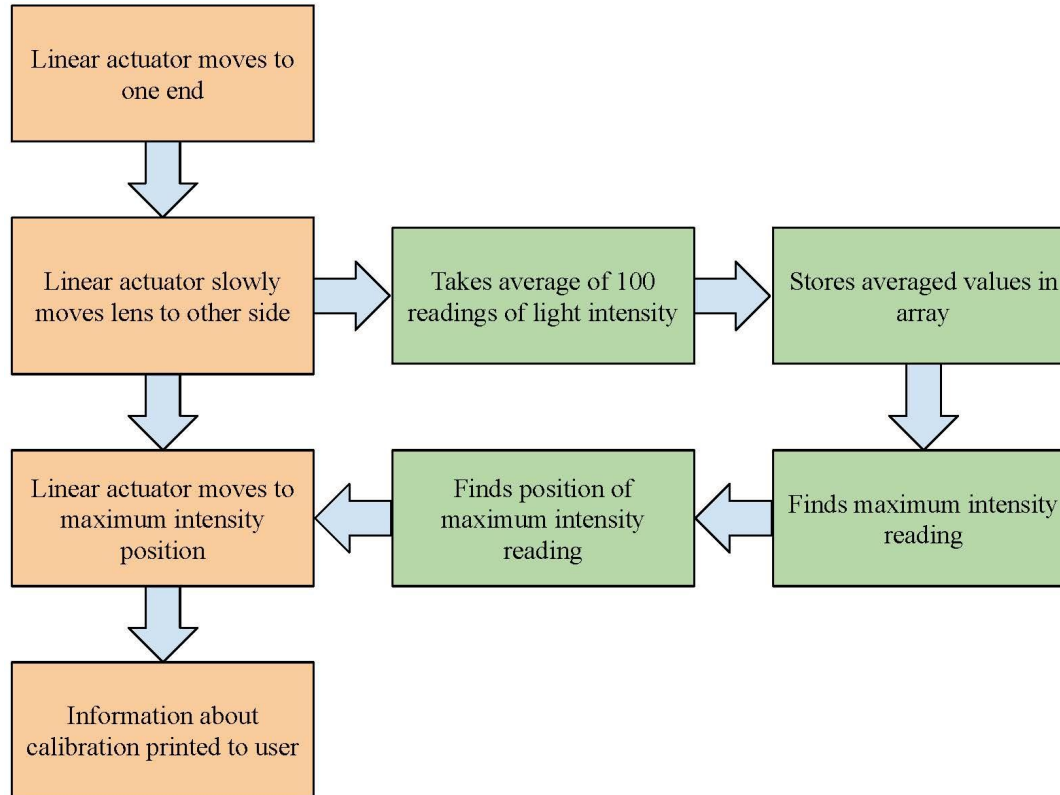
- Arduino Uno
- Photo-resistor
- Befenybay 200mm Linear Stage Actuator
- A4988 Stepper Motor Driver
- Convex Focusing Lens
- 5V Power Supply
- 12V Power Supply
- Lens Mount
- Liquid-Crystal Display
- 2x Limit Switch
- Breadboard
- 22 AWG Wire



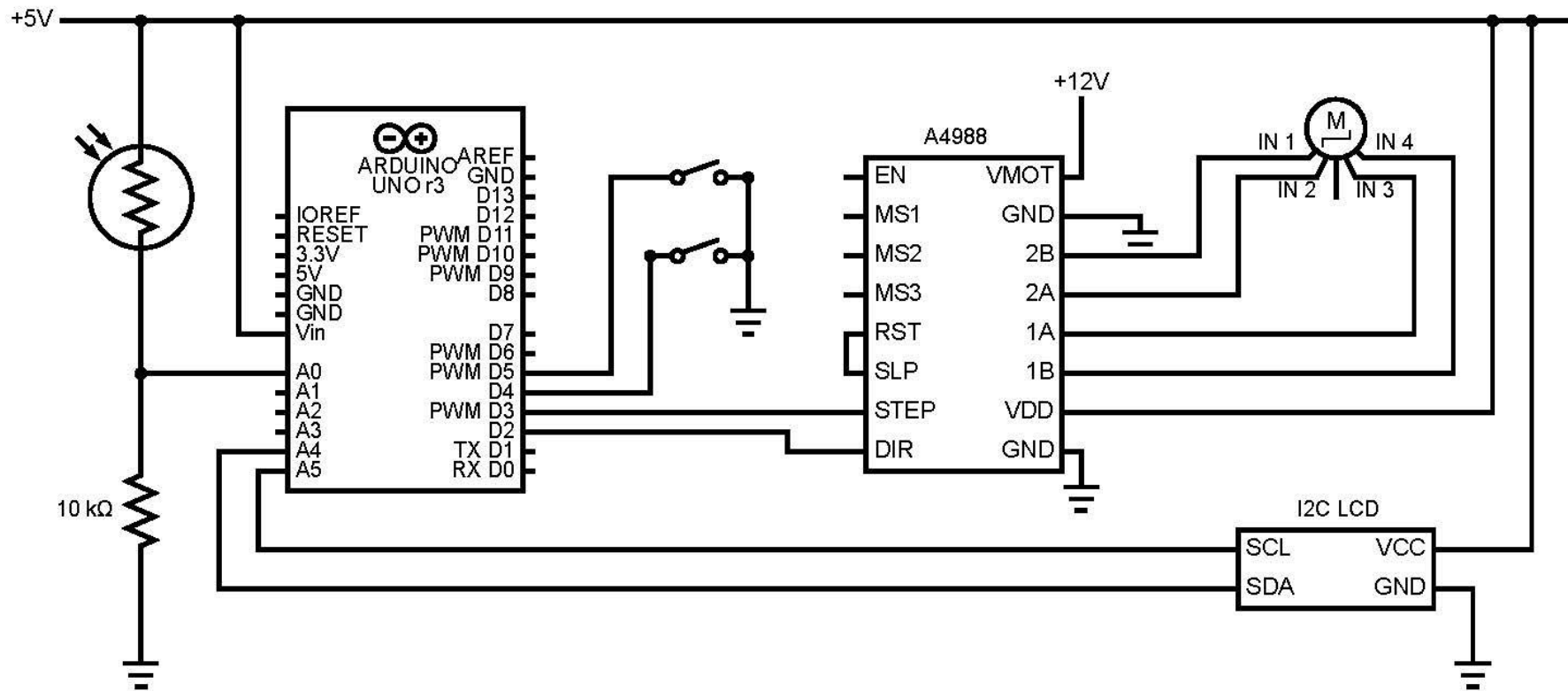
Hardware Block Diagram



Software Block Diagram



Schematic



Core Code Functions

```
int takereading()
{
    // record 100 readings into array
    int reading_array [100];
    for (int i=0;i<100;i++)
    {
        reading_array[i] = analogRead(A0);
        delay(1);
    }
    // find average reading

    long reading = 0;
    for (int i=0;i<100;i++)
    {
        reading=reading + reading_array[i];
    }
    reading=reading/100;
    // print average value
    return reading;
}
```

```
void movemotor(int steps,int direction,int speed)
{
    int dirPin = 2;
    int stepPin = 3;
    pinMode(stepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);

    //set direction
    digitalWrite(dirPin, direction);

    //go!
    for(int x = 0; x < steps; x++)
    {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(speed);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(speed);
    }
}
```

```
int checkswitch()
{
    // Check if the limit switch
    if (digitalRead(4) == LOW) {
        //Serial.println("Forward limit switch pressed");
        return 4;
    }

    // Check if the limit switch
    if (digitalRead(5) == LOW) {
        //Serial.println("Backward limit switch pressed");
        return 5;
    }
    else
    {
        return 0;
    }
}
```