

Project Name:

Motion following motorized camera base.

Project Description:

Our project was to make a camera base that can cover around 180 degrees and also can move according to the motion detected.

For the motion detection, we used PIR motion sensors and we used five of them to cover up 180 degrees. To move the camera base, we used servo motor. What our project does is, detecting motions with the help of PIRs and getting the angle, then rotates the servo to that direction. There are also 5 LED "status" lights positioned inside the base that will light up anytime one of the PIR sensors detects motion.

We extended our project to take snapshots too. Well, we tried to use java interface for taking snaps when any motion is detected. But, we couldn't do so. So, what we did is take snaps after five seconds while our camera base is working and save them to a folder. We used Java interface to do so.

Required Hardware:

1. Arduino UNO (1)
2. Capacitor (1)
3. LED (5)
4. PIR Motion Sensor (5)
5. Resistor (5)
6. Servo motor (1)
7. Bread board
8. Jumper wires

Required Software:

1. Arduino IDE
2. Netbeans IDE

How the code works:

```
void loop() {  
  
    for (int PIR = 0; PIR < 5; PIR++) { // start this loop for each PIR sensor  
        currentPIRpin = PIRpin[PIR]; // set current PIR pin to current number in 'for'  
loop  
        currentLEDpin=LEDpin[PIR]; // set current LED pin to current number in 'for' loop  
        PIRstatus = digitalRead(currentPIRpin);  
  
        if (PIRstatus == HIGH) { // if motion is detected on current PIR sensor  
            digitalWrite(currentLEDpin, HIGH); // turn corresponding LED on  
            if(PIRprevState[PIR] == 0) { // if PIR sensor's previous state is LOW  
                if (currentPIRposition != currentPIRpin && PIRprevState[PIR] == 0) { // if  
high PIR is different than current position PIR then move to new position  
                    camServo.write(PIRposition[PIR]);  
                    Serial.print("Current angle : ");  
                    Serial.println(PIRposition[PIR]);  
                    delay(50);  
                    currentPIRposition = currentPIRpin; // reset current PIR position to active  
[PIR] pin  
                    PIRprevState[PIR] = 1; // set previous PIR state to HIGH  
                }  
                PIRprevState[PIR] = 1; // set previous PIR state to HIGH if the current  
position is the same as the current PIR pin  
            } // end PIRprevState if  
            } // end PIRstatus if  
  
            else { //  
                digitalWrite(currentLEDpin, LOW); //the led visualizes the sensors output pin  
state  
                PIRprevState[PIR] = 0; // set previous PIR state to LOW  
            } // end else  
  
        } // end [PIR] for loop  
    } // end main loop  
}
```

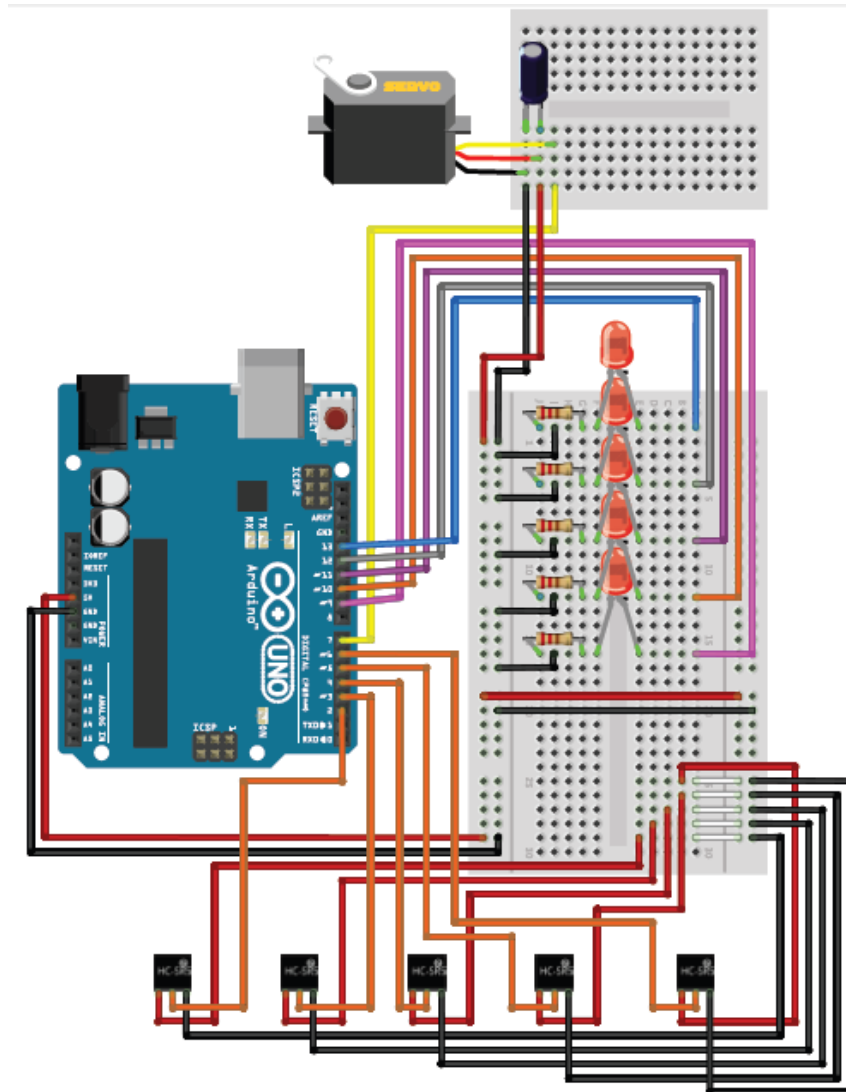
As you see, we managed all the functionalities in the loop function of Arduino . We did not need/add any additional functions for our project.

The Java Interface is given below:

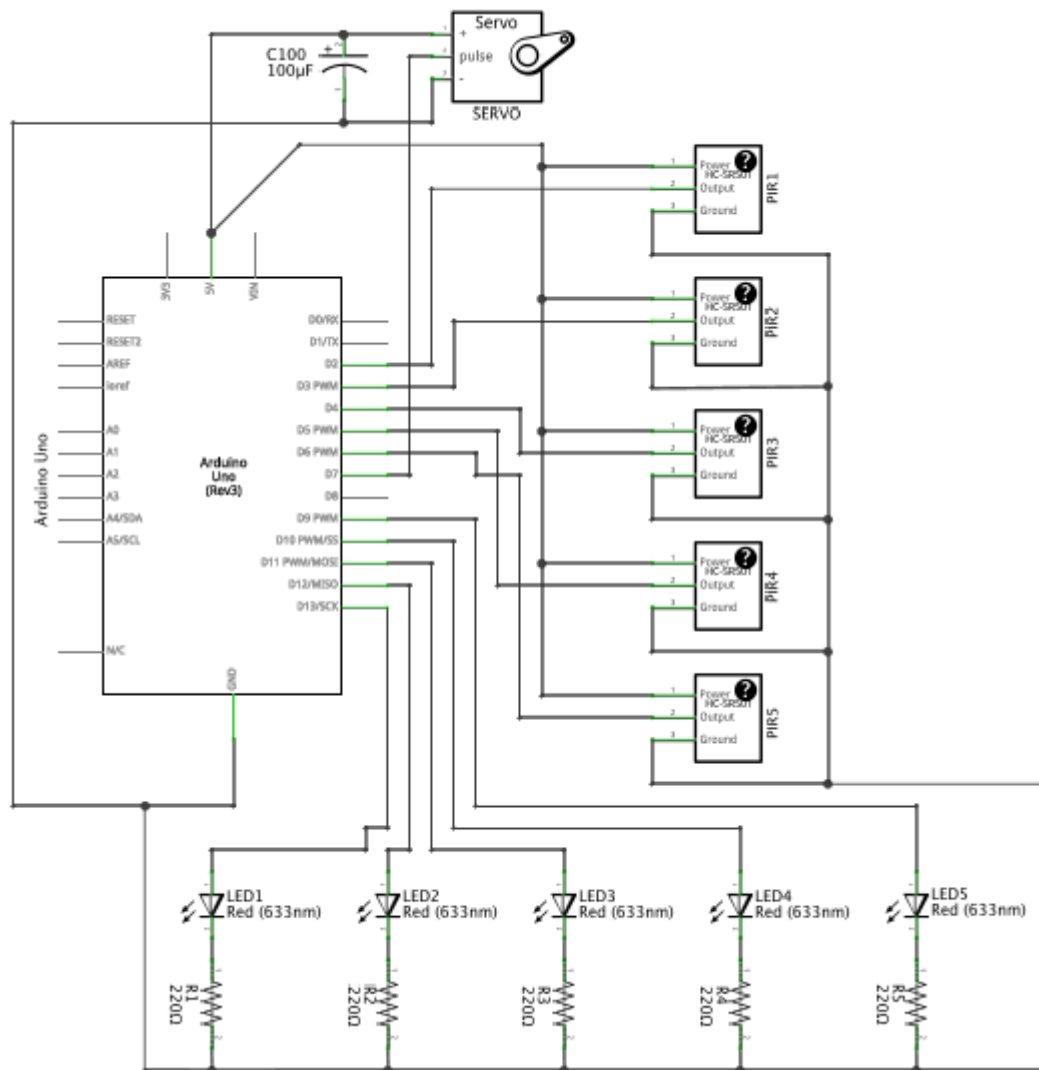
```
1. public class GrabberShow implements Runnable {
2.     //final int INTERVAL=1000;///you may use interval
3.     IplImage image;
4.     CanvasFrame canvas = new CanvasFrame("Web Cam");
5.     public GrabberShow() {
6.         canvas.setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);
7.     }
8.     @Override
9.     public void run() {
10.        FrameGrabber grabber = new VideoInputFrameGrabber(0);
11.        int i=0;
12.        try {
13.            grabber.start();
14.            IplImage img;
15.            while (true) {
16.                img = grabber.grab();
17.                if (img != null) {
18.                    cvFlip(img, img, 1);// l-r = 90_degrees_steps_anti_clockwise
19.                    cvSaveImage((i++)+"-capture.jpg", img);
20.                    // show image on window
21.                    canvas.showImage(img);
22.                }
23.                //Thread.sleep(INTERVAL);
24.            }
25.        } catch (Exception e) {
26.        }
27.    }
```

This is where we interfaced the webcam. And it takes snaps after 5 seconds.

The Circuit Diagram:



The Schematic Diagram:



Experience:

It was totally a new experience for us as we never worked with Arduino before. We learnt so many things and we could have done better. Nevertheless, it was a good experience for three of us.