

VIT-AP
UNIVERSITY

ECS1001 – LED Bike Vest

Guided by –
Dr. Hari Kishan Kondaveeti



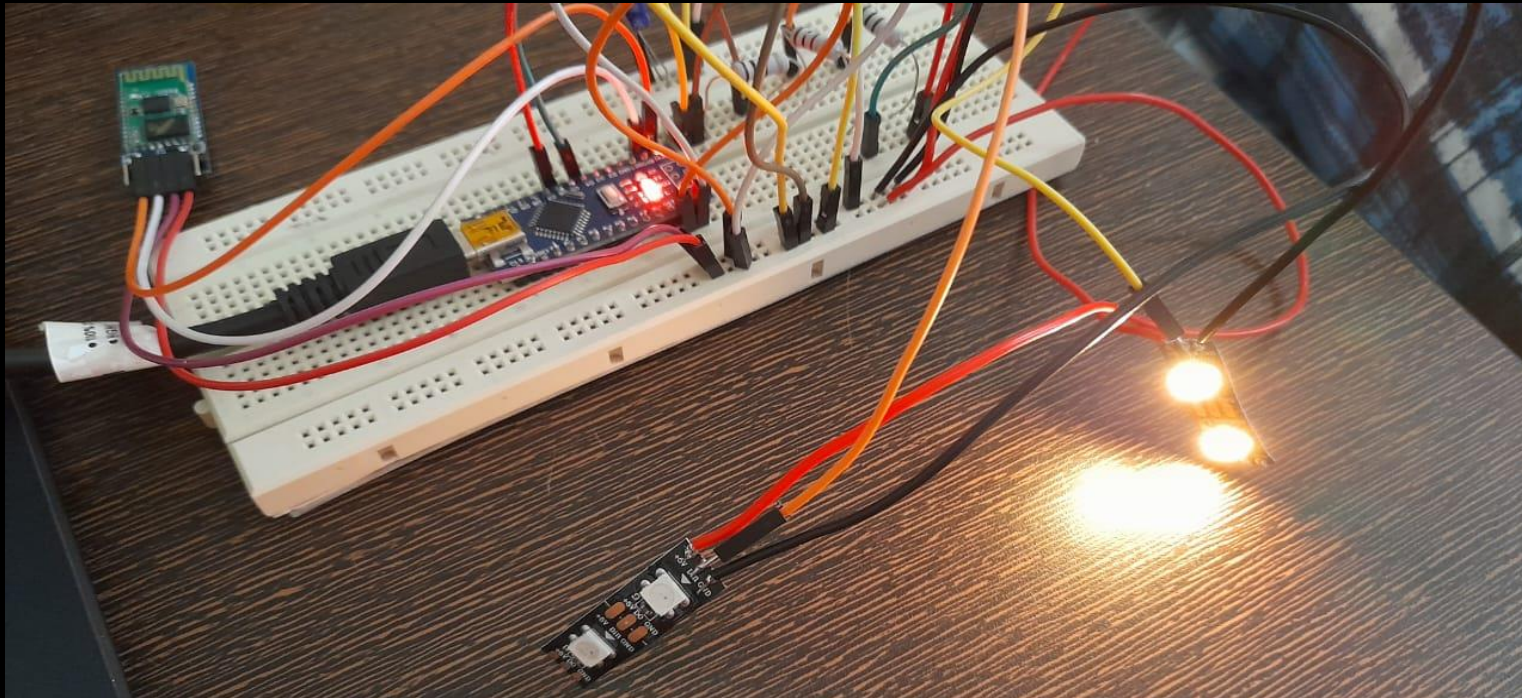
Team Members

- ◊ NIRDESH SINGH 20BCE7062
- ◊ VARIGONDA ANJANI GAYATHRI
20BCI7003
- ◊ MANYA ARORA 20BCE7441
- ◊ SAKETI BHAVANI 20BES7073
- ◊ DEVARAPU KUSUMA 20BEC7074
- ◊ ALAGAPPAN SP 20BCE7211

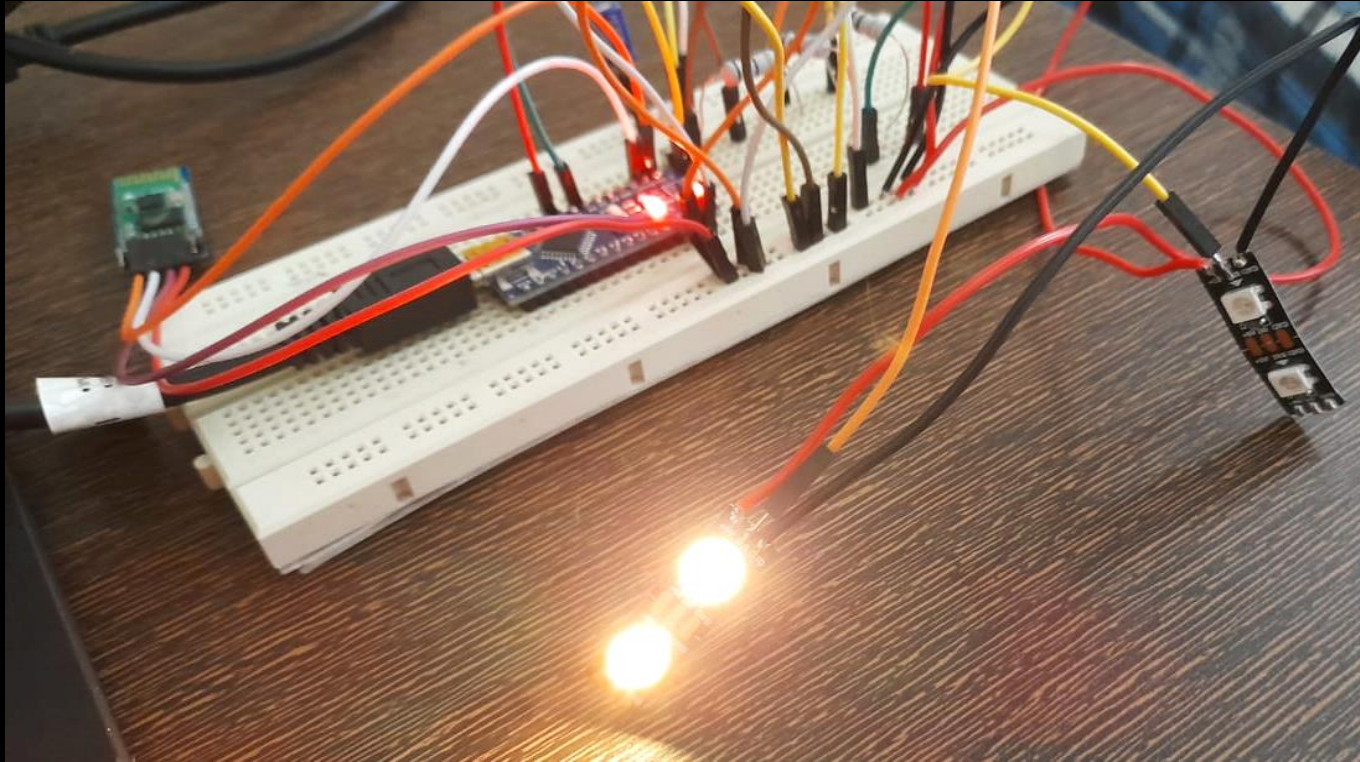
Circuit connections

◇ VEST

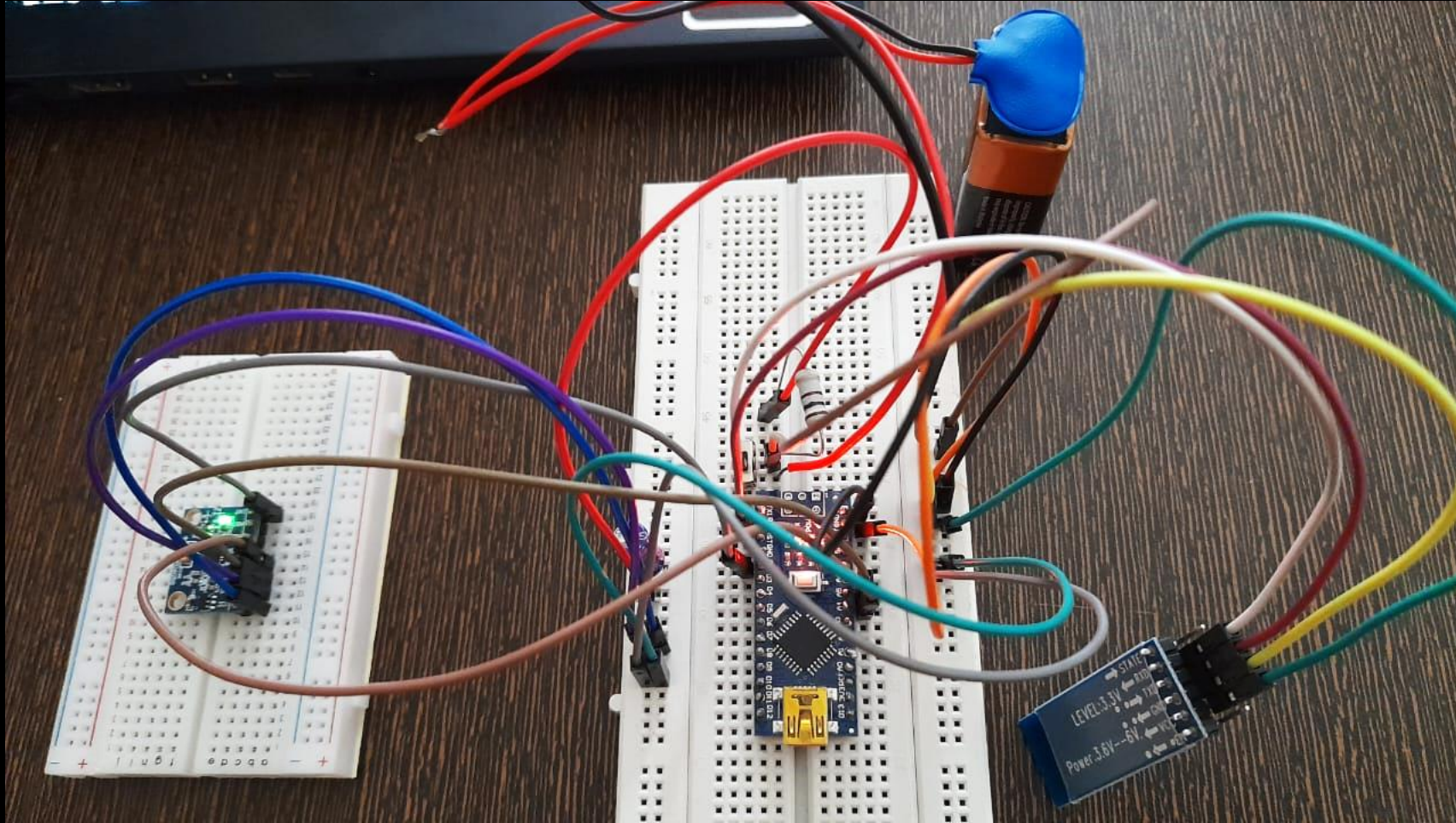
Circuit indicating the blinking of right LEDs



Circuit indicating the blinking of left LEDs



Circuit of the remote



The following is code for LED vest circuit

sketch_nov19a.ino

```
1  #include <Adafruit_NeoPixel.h>
2  #ifdef _AVR_
3  #include <avr/power.h>
4  #endif
5
6  #define LED_LEFT    4
7  #define LED_RIGHT   6
8
9  #define LED_COUNT 4 // Define the number of LEDs in the strip
10
11  char state = 0;
12  int light_delay = 50;
13
14  Adafruit_NeoPixel strip_left(LED_COUNT, LED_LEFT, NEO_GRB + NEO_KHZ800);
15  Adafruit_NeoPixel strip_right(LED_COUNT, LED_RIGHT, NEO_GRB + NEO_KHZ800);
16
17  void setup() {
18
19  #if defined(_AVR_ATtiny85_) && (F_CPU == 16000000)
20  |   clock_prescale_set(clock_div_1);
21  #endif
22
23  strip_left.begin();
24  strip_left.show();
25  strip_left.setBrightness(150);
26
```

sketch_nov19a.ino

```
26
27   strip_right.begin();
28   strip_right.show();
29   strip_right.setBrightness(150);
30   Serial.begin(9600);
31   delay(1000);
32 }
33
34 void loop() {
35
36   if (Serial.available() > 0) {
37
38
39     state = Serial.read();
40     Serial.print(state);
41
42     if (state == 'L') {
43       leftBlink();
44       delay(light_delay);
45     }
46
47
48     if (state == 'R') {
49       rightBlink();
50       delay(light_delay);
51     }
52 }
```


sketch_nov19a.ino

```
52     }
53
54 }
55
56 }
57 void colorWipeLeft(uint32_t color, int wait) {
58     for (int i = 0; i < strip_left.numPixels(); i++) {
59         strip_left.setPixelColor(i, color);
60         strip_left.show();
61         delay(wait);
62         strip_left.clear();
63     }
64 }
65
66
67 void rightBlink() {
68
69     for (int i = 0; i < 4; i++) {
70
71         colorBlinkersRight(strip_right.Color(255, 100, 0), 50); //Yellow
72         delay(400);
73         colorBlinkersRight(strip_right.Color(0, 0, 0), 25); //Yellow
74         delay(300);
75     }
76 }
77 }
```



```
79 void colorBlinkersRight(uint32_t c, int wait) {
80
81     for(int i=0;i<2;i++) {
82         strip_right.setPixelColor(i, c);
83     }
84
85     strip_right.show();
86     delay(wait);
87     strip_right.clear();
88     delay(wait);
89
90 }
91
92 void hazardBlink() {
93
94     for (int i = 0; i < 4; i++) {
95
96         colorBlinkersLeft(strip_left.Color(255, 100, 0), 50); //Yellow
97         colorBlinkersRight(strip_right.Color(255, 100, 0), 50); //Yellow
98
99         delay(400);
100         colorBlinkersLeft(strip_left.Color(0, 0, 0), 25); //Yellow
101         colorBlinkersRight(strip_right.Color(0, 0, 0), 50); //Yellow
102
103         delay(300);
104
105     }
```

```

106 }
107
108
109 void leftBlink() {
110
111     for (int i = 0; i < 4; i++) {
112
113         colorBlinkersLeft(strip_left.Color(255, 100, 0), 50); //Yellow
114         delay(400);
115         colorBlinkersLeft(strip_left.Color(0, 0, 0), 25); //Yellow
116         delay(300);
117     }
118 }
119
120
121 void colorBlinkersLeft(uint32_t c, int wait) {
122
123     for(int i=0;i<2;i++) {
124         strip_left.setPixelColor(i, c);
125     }
126
127
128     strip_left.show();
129     delay(wait);
130     strip_left.clear();
131     delay(wait);
132

```

```
133 }
134
135 void colorWipeRight(uint32_t color, int wait) {
136     for (int i = 0; i < strip_right.numPixels(); i++) {
137         strip_right.setPixelColor(i, color);
138         strip_right.show();
139         delay(wait);
140         strip_right.clear();
141     }
142 }
143
144 void rainbow_left(int wait) {
145
146     for (long firstPixelHue = 0; firstPixelHue < 5 * 65536; firstPixelHue += 256) {
147         for (int i = 0; i < strip_left.numPixels(); i++) {
148
149             int pixelHue = firstPixelHue + (i * 65536L / strip_left.numPixels());
150
151             strip_left.setPixelColor(i, strip_left.gamma32(strip_left.ColorHSV(pixelHue)));
152         }
153         strip_left.show();
154         delay(wait); // Pause for a moment
155     }
156 }
157
158 void rainbow_right(int wait) {
159
```

```
156 }
157
158 void rainbow_right(int wait) {
159
160     for (long firstPixelHue = 0; firstPixelHue < 5 * 65536; firstPixelHue += 256) {
161         for (int i = 0; i < strip_right.numPixels(); i++) {
162
163             int pixelHue = firstPixelHue + (i * 65536L / strip_right.numPixels());
164
165             strip_right.setPixelColor(i, strip_right.gamma32(strip_right.ColorHSV(pixelHue)));
166         }
167         strip_right.show();
168         delay(wait); // Pause for a moment
169     }
170 }
171
172 void clearLights() {
173     strip_left.clear();
174     strip_right.clear();
175 }
```


The following is the code for the handle circuit

sketch_nov19b.ino

```
1  #include<Wire.h>
2
3  const int MPU_addr = 0x68;
4  int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
5
6  int minVal = 265;
7  int maxVal = 402;
8
9  double x;
10 double y;
11 double z;
12
13 bool bool_caliberate = false;
14 int response_time = 400;
15
16 bool offset_pos = false;
17
18 void setup() {
19
20     Wire.begin();
21     Wire.beginTransmission(MPU_addr);
22     Wire.write(0x6B);
23     Wire.write(0);
24     Wire.endTransmission(true); Wire.begin();
25     Serial.begin(9600);
26     delay(1000);
27
28 }
```

```
34
35     if (offset_pos == false) {
36         if ( GyZ > 15000) {
37             Serial.println("L");
38             offset_pos = true;
39             delay(response_time);
40         }
41     }
42
43     if (offset_pos == true) {
44         if ( GyZ > 15000) {
45             Serial.println("L");
46             offset_pos = false;
47             delay(response_time);
48         }
49     }
50
51     if (offset_pos == true) {
52         if ( GyZ < -15000) {
53             Serial.println("R");
54             delay(response_time);
55             offset_pos = false;
56         }
57     }
58 }
59
60 if (offset_pos == false) {
61     if ( GyZ < -15000) {
62         Serial.println("R");
63         delay(response_time);
64         offset_pos = true;
65     }
66 }
67 }
68
69 }
```

```
69 }
70
71 void GetMpuValue1(const int MPU) {
72
73     Wire.beginTransmission(MPU);
74     Wire.write(0x3B);
75     Wire.endTransmission(false);
76     Wire.requestFrom(MPU, 14, true);
77
78     AcX = Wire.read() << 8 | Wire.read();
79     AcY = Wire.read() << 8 | Wire.read();
80     AcZ = Wire.read() << 8 | Wire.read();
81
82     Tmp = Wire.read() << 8 | Wire.read();
83
84     int xAng = map(AcX, minVal, maxVal, -90, 90);
85     int yAng = map(AcY, minVal, maxVal, -90, 90);
86     int zAng = map(AcZ, minVal, maxVal, -90, 90);
87
88     GyX = Wire.read() << 8 | Wire.read();
89     GyY = Wire.read() << 8 | Wire.read();
90     GyZ = Wire.read() << 8 | Wire.read();
91     x = RAD_TO_DEG * (atan2(-yAng, -zAng) + PI) + 4;
92     y = RAD_TO_DEG * (atan2(-xAng, -zAng) + PI);
93     z = RAD_TO_DEG * (atan2(-yAng, -xAng) + PI);
94
95 }
96
97 void MPU_debug() {
98
99 }
```

Output of the following prototype

```
Output  Serial Monitor X
Message (Enter to send message to 'Arduino Nano' on 'COM9')
R
L
R
L
R
L
L
L
R
L
```