RELAY CODING

```
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
pinMode(2,INPUT);
pinMode(3,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.println(digitalRead(2));
  if(digitalRead(2)==1)
{
  for(int i=0;i<60;i++)
  {
    digitalWrite(3,HIGH);
    delay(1000);
  }
}
else if(digitalRead(2)==0)
{
  digitalWrite(3,LOW);   }
```

WELCOME TO LIGHTS OUT – CODING

```
// defining the alphabet

const unsigned char font[95][5] = {

 {

   0x00,0x00,0x00,0x00,0x00      }

 ,  //   0x20 32

 {

   0x00,0x00,0x6f,0x00,0x00      }

 ,  // ! 0x21 33

 {

   0x00,0x07,0x00,0x07,0x00      }

 ,  // " 0x22 34

 {

   0x14,0x7f,0x14,0x7f,0x14      }

 ,  // # 0x23 35

 {

   0x00,0x07,0x04,0x1e,0x00      }

 ,  // $ 0x24 36

 {

   0x23,0x13,0x08,0x64,0x62      }

 ,  // % 0x25 37

 {

   0x36,0x49,0x56,0x20,0x50      }

 ,  // & 0x26 38

 {
   0x00,0x00,0x07,0x00,0x00      }
 ,   // ' 0x27 39
 {
   0x00,0x1c,0x22,0x41,0x00      }
 ,   // ( 0x28 40
 {
   0x00,0x41,0x22,0x1c,0x00      }
 ,   // ) 0x29 41
 {
   0x14,0x08,0x3e,0x08,0x14      }
 ,   // * 0x2a 42
 {
   0x08,0x08,0x3e,0x08,0x08      }
 ,   // + 0x2b 43
 {
   0x00,0x50,0x30,0x00,0x00      }
 ,   // , 0x2c 44
 {
   0x08,0x08,0x08,0x08,0x08      }
 ,   // - 0x2d 45
 {
   0x00,0x60,0x60,0x00,0x00      }
 ,   // . 0x2e 46
 {
   0x20,0x10,0x08,0x04,0x02      }
```

```
,   // / 0x2f 47
{
  0x3e,0x51,0x49,0x45,0x3e      }
,   // 0 0x30 48
{
  0x00,0x42,0x7f,0x40,0x00      }
,   // 1 0x31 49
{
  0x42,0x61,0x51,0x49,0x46      }
,   // 2 0x32 50
{
  0x21,0x41,0x45,0x4b,0x31      }
,   // 3 0x33 51
{
  0x18,0x14,0x12,0x7f,0x10      }
,   // 4 0x34 52
{
  0x27,0x45,0x45,0x45,0x39      }
,   // 5 0x35 53
{
  0x3c,0x4a,0x49,0x49,0x30      }
,   // 6 0x36 54
{
  0x01,0x71,0x09,0x05,0x03      }
,   // 7 0x37 55
```

```
{
  0x36,0x49,0x49,0x49,0x36      }
,   // 8 0x38 56
{
  0x06,0x49,0x49,0x29,0x1e      }
,   // 9 0x39 57
{
  0x00,0x36,0x36,0x00,0x00      }
,   // : 0x3a 58
{
  0x00,0x56,0x36,0x00,0x00      }
,   // ; 0x3b 59
{
  0x08,0x14,0x22,0x41,0x00      }
,   // < 0x3c 60
{
  0x14,0x14,0x14,0x14,0x14      }
,   // = 0x3d 61
{
  0x00,0x41,0x22,0x14,0x08      }
,   // > 0x3e 62
{
  0x02,0x01,0x51,0x09,0x06      }
,   // ? 0x3f 63
{
  0x3e,0x41,0x5d,0x49,0x4e      }
,   // @ 0x40 64
{
  0x7e,0x09,0x09,0x09,0x7e      }
,   // A 0x41 65
{
  0x7f,0x49,0x49,0x49,0x36      }
,   // B 0x42 66
{
  0x3e,0x41,0x41,0x41,0x22      }
,   // C 0x43 67
{
  0x7f,0x41,0x41,0x41,0x3e      }
,   // D 0x44 68
{
  0x7f,0x49,0x49,0x49,0x41      }
,   // E 0x45 69
{
  0x7f,0x09,0x09,0x09,0x01      }
,   // F 0x46 70
{
  0x3e,0x41,0x49,0x49,0x7a      }
,   // G 0x47 71
{
  0x7f,0x08,0x08,0x08,0x7f      }
```

```
,   // H 0x48 72
{
  0x00,0x41,0x7f,0x41,0x00      }
,   // I 0x49 73
{
  0x20,0x40,0x41,0x3f,0x01      }
,   // J 0x4a 74
{
  0x7f,0x08,0x14,0x22,0x41      }
,   // K 0x4b 75
{
  0x7f,0x40,0x40,0x40,0x40      }
,   // L 0x4c 76
{
  0x7f,0x02,0x0c,0x02,0x7f      }
,   // M 0x4d 77
{
  0x7f,0x04,0x08,0x10,0x7f      }
,   // N 0x4e 78
{
  0x3e,0x41,0x41,0x41,0x3e      }
,   // O 0x4f 79
{
  0x7f,0x09,0x09,0x09,0x06      }
,   // P 0x50 80
```

```
{
  0x3e,0x41,0x51,0x21,0x5e      }
,   // Q 0x51 81
{
  0x7f,0x09,0x19,0x29,0x46      }
,   // R 0x52 82
{
  0x46,0x49,0x49,0x49,0x31      }
,   // S 0x53 83
{
  0x01,0x01,0x7f,0x01,0x01      }
,   // T 0x54 84
{
  0x3f,0x40,0x40,0x40,0x3f      }
,   // U 0x55 85
{
  0x0f,0x30,0x40,0x30,0x0f      }
,   // V 0x56 86
{
  0x3f,0x40,0x30,0x40,0x3f      }
,   // W 0x57 87
{
  0x63,0x14,0x08,0x14,0x63      }
,   // X 0x58 88
{
  0x07,0x08,0x70,0x08,0x07      }

,   // Y 0x59 89

{

  0x61,0x51,0x49,0x45,0x43      }

,   // Z 0x5a 90

{

  0x3c,0x4a,0x49,0x29,0x1e      }

,   // [ 0x5b 91

{

  0x02,0x04,0x08,0x10,0x20      }

,   // \ 0x5c 92

{

  0x00,0x41,0x7f,0x00,0x00      }

,   // ] 0x5d 93

{

  0x04,0x02,0x01,0x02,0x04      }

,   // ^ 0x5e 94

{

  0x40,0x40,0x40,0x40,0x40      }

,   // _ 0x5f 95

{

  0x00,0x00,0x03,0x04,0x00      }

,   // ` 0x60 96

{

  0x20,0x54,0x54,0x54,0x78      }
```

```
,   // a 0x61 97

{

  0x7f,0x48,0x44,0x44,0x38      }

,   // b 0x62 98

{

  0x38,0x44,0x44,0x44,0x20      }

,   // c 0x63 99

{

  0x38,0x44,0x44,0x48,0x7f      }

,   // d 0x64 100

{

  0x38,0x54,0x54,0x54,0x18      }

,   // e 0x65 101

{

  0x08,0x7e,0x09,0x01,0x02      }

,   // f 0x66 102

{

  0x0c,0x52,0x52,0x52,0x3e      }

,   // g 0x67 103

{

  0x7f,0x08,0x04,0x04,0x78      }

,   // h 0x68 104

{

  0x00,0x44,0x7d,0x40,0x00      }

,   // i 0x69 105
```

```
{
  0x20,0x40,0x44,0x3d,0x00      }
,   // j 0x6a 106
{
  0x00,0x7f,0x10,0x28,0x44      }
,   // k 0x6b 107
{
  0x00,0x41,0x7f,0x40,0x00      }
,   // l 0x6c 108
{
  0x7c,0x04,0x18,0x04,0x78      }
,   // m 0x6d 109
{
  0x7c,0x08,0x04,0x04,0x78      }
,   // n 0x6e 110
{
  0x38,0x44,0x44,0x44,0x38      }
,   // o 0x6f 111
{
  0x7c,0x14,0x14,0x14,0x08      }
,   // p 0x70 112
{
  0x08,0x14,0x14,0x18,0x7c      }
,   // q 0x71 113
{
  0x7c,0x08,0x04,0x04,0x08      }
,   // r 0x72 114
{
  0x48,0x54,0x54,0x54,0x20      }
,   // s 0x73 115
{
  0x04,0x3f,0x44,0x40,0x20      }
,   // t 0x74 116
{
  0x3c,0x40,0x40,0x20,0x7c      }
,   // u 0x75 117
{
  0x1c,0x20,0x40,0x20,0x1c      }
,   // v 0x76 118
{
  0x3c,0x40,0x30,0x40,0x3c      }
,   // w 0x77 119
{
  0x44,0x28,0x10,0x28,0x44      }
,   // x 0x78 120
{
  0x0c,0x50,0x50,0x50,0x3c      }
,   // y 0x79 121
{
  0x44,0x64,0x54,0x4c,0x44      }
  ,   // z 0x7a 122
  {
    0x00,0x08,0x36,0x41,0x41      }
  ,   // { 0x7b 123
  {
    0x00,0x00,0x7f,0x00,0x00      }
  ,   // | 0x7c 124
  {
    0x41,0x41,0x36,0x08,0x00      }
  ,   // } 0x7d 125
  {
    0x04,0x02,0x04,0x08,0x04      }
  ,   // ~ 0x7e 126
};
/*
The LEDs on AltSense's POV shield are arranged in the following order
 starting from top...
 {11,12,13,14,15,19,18,17,16,8,1,0,2,3,10,9,7,6,5,4}
 */
// only first 7 LEDs are used for Hello World
const int LEDpins[] = {
  11,12,13,14,15,19,18
};
// number of LEDs
const int charHeight = sizeof(LEDpins);
const int charWidth = 5;

int columnDelay = 600;

int letterDelay = 1500;

void setup()

{

  for (int i = 0; i < 20; i++)

    pinMode(i, OUTPUT);

}


void loop()

{

  const char textString[] = "WELCOME TO LIGHTS OUT";

  for (int k=0; k<sizeof(textString); k++)

  {

    printLetter(textString[k]);

  }

  delay(112);

}

void printLetter(char ch)

{

  // make sure the character is within the alphabet bounds (defined by the font.h file)

  // if it's not, make it a blank character

  if (ch < 32 || ch > 126) {

    ch = 32;

  }
```

```
 // subtract the space character (converts the ASCII number to the font index number)
 ch -= 32;
 // step through each byte of the character array
 for (int i = 0; i < charWidth; i++) {
   byte b = font[ch][i];


   // bit shift through the byte and output it to the pin
   for (int j = 0; j < charHeight; j++) {
     digitalWrite(LEDpins[j], !!(b & (1 << j)));
   }
   // space between columns
delayMicroseconds(columnDelay);
 }
 //clear the LEDs
 digitalWrite(11 , LOW);   // set the LED on
 digitalWrite(12 , LOW);   // set the LED on
 digitalWrite(13 , LOW);   // set the LED on
 digitalWrite(14 , LOW);   // set the LED on
 digitalWrite(15 , LOW);  // set the LED on
 digitalWrite(19 , LOW);   // set the LED on
 digitalWrite(18 , LOW);   // set the LED on
 // space between letters
 delayMicroseconds(letterDelay);


}
```