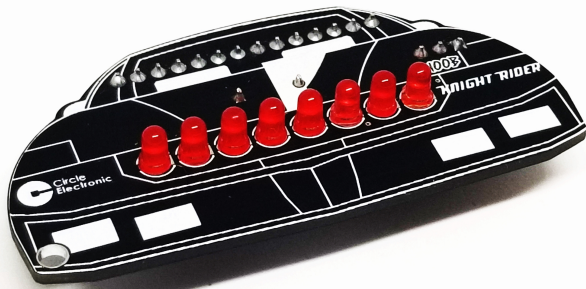
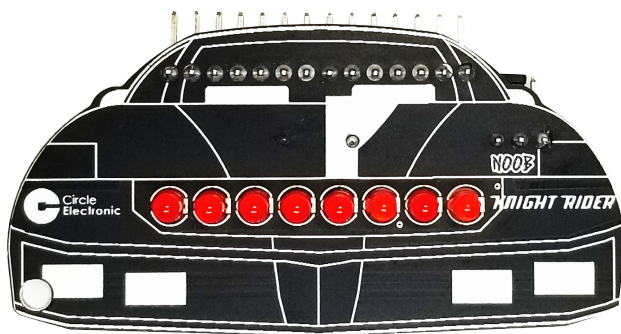




Knight Rider

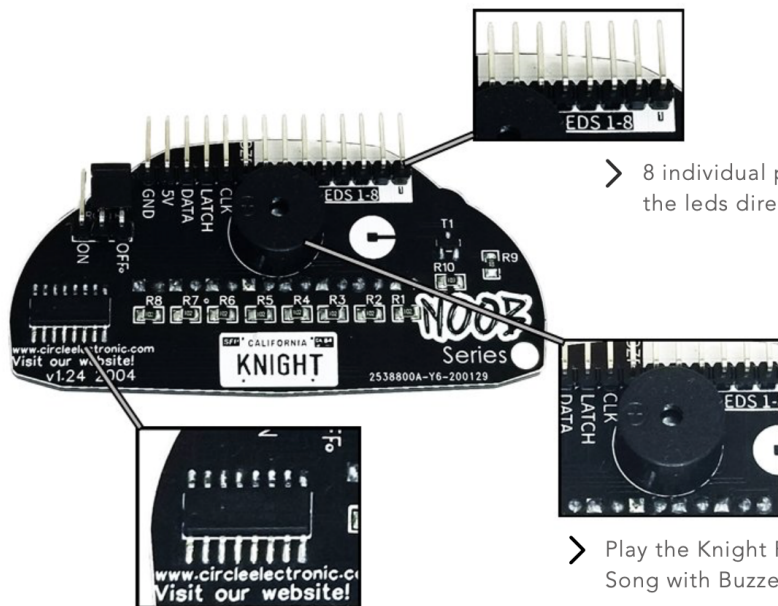


Every beginner learns to code knight rider led animation in first steps. If you have passion for code to knight rider led and you don't have time to build circuit for it, this product exactly for you.



NOOB Series Knight Rider has 1 buzzer and 8 LED's. You can manage LED's with 74hc595 shift register or individually. With this register, we can use the leds with 3 pins.

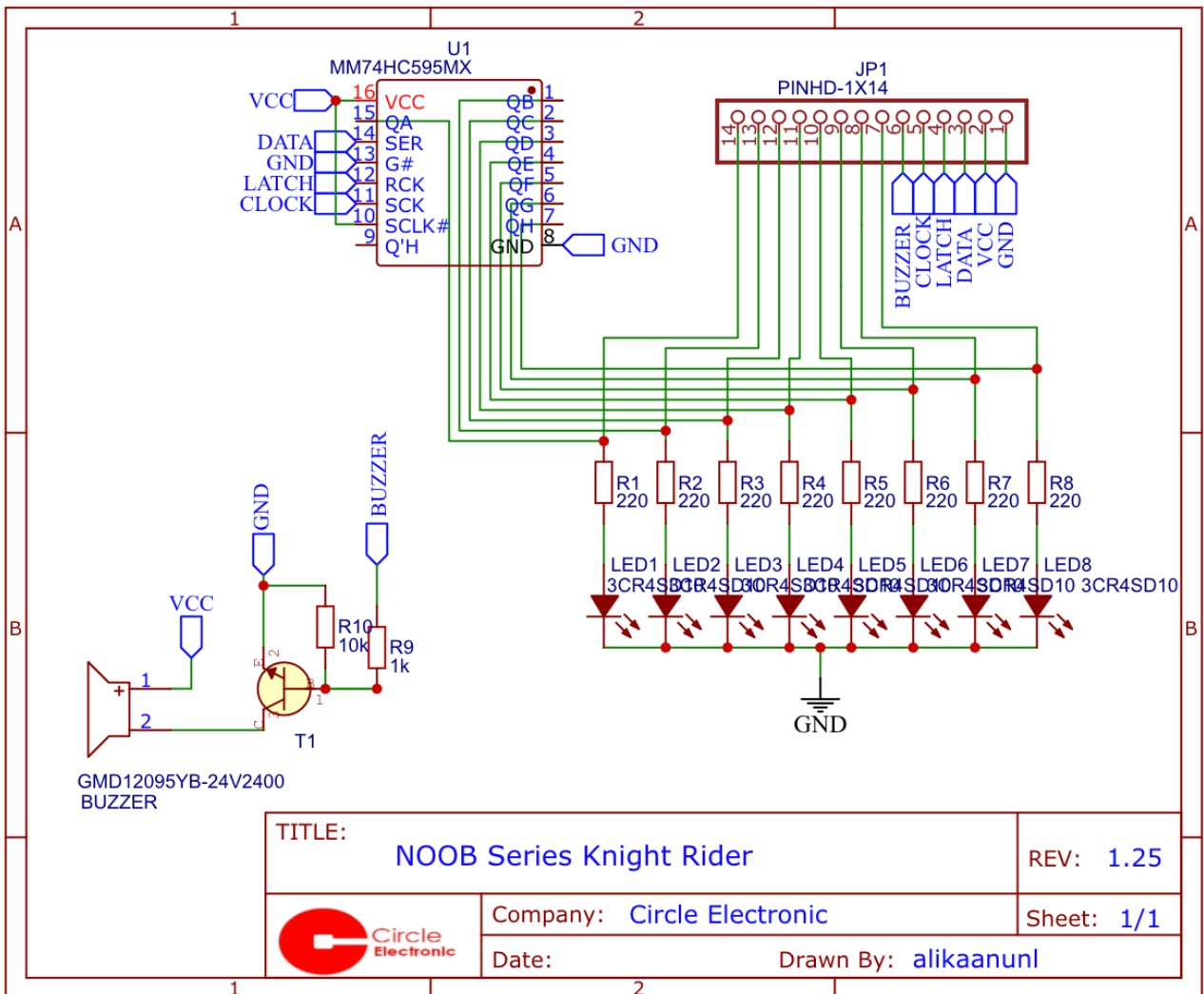
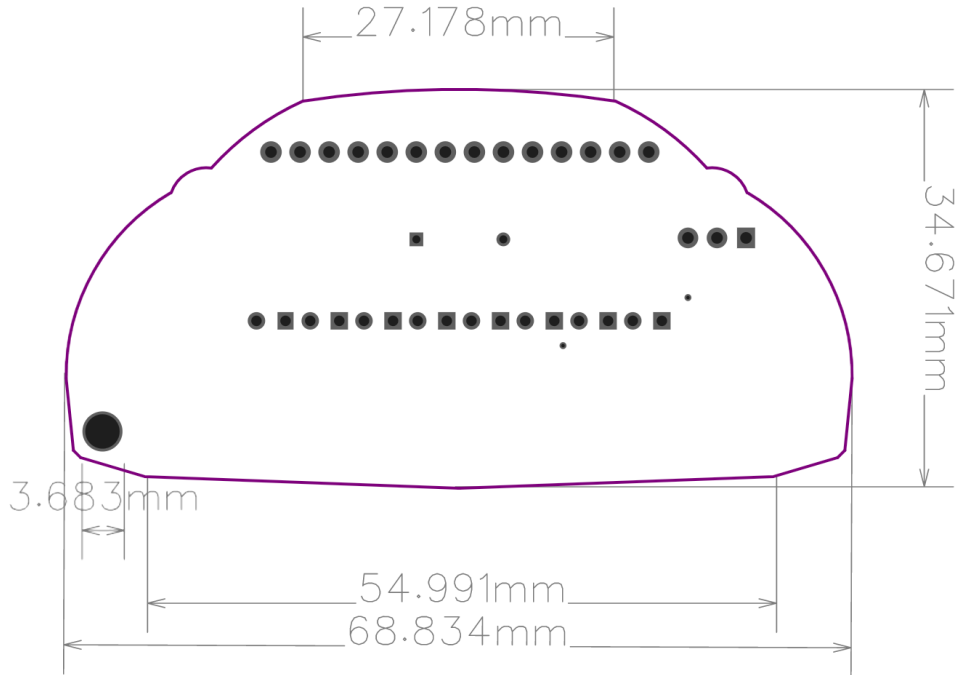
You can play the knight rider music with Buzzer. Buzzer working with transistor. With transistor we are protecting Arduino pin to high current.



> 8 individual pins for each led. Code the leds directly.

> Play the Knight Rider Song with Buzzer

> Use the leds with 74hc595 shift register. You can use the leds with 3 pin in this way.



TITLE: NOOB Series Knight Rider		REV: 1.25
	Company: Circle Electronic	Sheet: 1/1
	Date:	Drawn By: alikaanunl



Learn how to code

8 Pin Configuration

*v1.25 or latest do not use 5v pin and gnd pin at 8 pin configuration.

```
int leds[] = {2,3,4,5,6,7,8,9};
void setup()
{
  for(int i=0; i<8; i++) {
    pinMode(leds[i], OUTPUT);
  }
}
void loop()
{
  for(int i=0; i<7; i++) {
    digitalWrite(leds[i], HIGH);
    delay(100);
    digitalWrite(leds[i], LOW);
  }
  for(int j=7; j>0; j--) {
    digitalWrite(leds[j], HIGH);
    delay(100);
    digitalWrite(leds[j], LOW);
  }
}
```

74hc595 Shift Register Configuration

```
#define LATCH 9
#define CLOCK 10
#define DATA 8
static int led = 0;
byte number[23] = {0b00000000,
  0b00000001,
  0b00000011,
  0b00000111,
  0b00001110,
  0b00011100,
  0b00111000,
  0b01110000,
  0b11100000,
  0b11000000,
  0b10000000,
  0b00000000,
  0b10000000,
  0b11000000,
  0b11100000,
  0b01110000,
  0b00111000,
  0b00011100,
  0b00001110,
  0b00000111,
  0b00000011,
  0b00000001,
```



```

        0b00000000
    };
void setup() {
    pinMode(CLOCK, OUTPUT);
    pinMode(DATA, OUTPUT);
    pinMode(LATCH, OUTPUT);
}
void loop() {
    static unsigned long time = millis();
    if (millis() - time >= 80 && led <= 22) {
        time = millis();
        led++;
        digitalWrite(LATCH, LOW);
        shiftOut(DATA, CLOCK, MSBFIRST, number[led]);
        digitalWrite(LATCH, HIGH);
    }
    if (led == 22) {
        led = 0;
    }
}

```

Knight Rider Music Configuration (For playing music we need to use millis function instead of delay function this method allows the processor to multitasking)

```

#define NOTE_B0 31 #define NOTE_C1 33 #define NOTE_CS1 35 #define NOTE_D1 37 #define NOTE_DS1 39
#define NOTE_E1 41 #define NOTE_F1 44 #define NOTE_FS1 46
#define NOTE_G1 49 #define NOTE_GS1 52 #define NOTE_A1 55 #define NOTE_AS1 58
#define NOTE_B1 62 #define NOTE_C2 65 #define NOTE_CS2 69 #define NOTE_D2 73
#define NOTE_DS2 78 #define NOTE_E2 82 #define NOTE_F2 87 #define NOTE_FS2 93
#define NOTE_G2 98 #define NOTE_GS2 104 #define NOTE_A2 110
#define NOTE_AS2 117 #define NOTE_B2 123 #define NOTE_C3 131
#define NOTE_CS3 139 #define NOTE_D3 147 #define NOTE_DS3 156
#define NOTE_E3 165 #define NOTE_F3 175 #define NOTE_FS3 185
#define NOTE_G3 196 #define NOTE_GS3 208 #define NOTE_A3 220
#define NOTE_AS3 233 #define NOTE_B3 247 #define NOTE_C4 262
#define NOTE_CS4 277 #define NOTE_D4 294 #define NOTE_DS4 311
#define NOTE_E4 330 #define NOTE_F4 349 #define NOTE_FS4 370
#define NOTE_G4 392 #define NOTE_GS4 415 #define NOTE_A4 440
#define NOTE_AS4 466 #define NOTE_B4 494 #define NOTE_C5 523
#define NOTE_CS5 554 #define NOTE_D5 587 #define NOTE_DS5 622
#define NOTE_E5 659 #define NOTE_F5 698 #define NOTE_FS5 740
#define NOTE_G5 784 #define NOTE_GS5 831 #define NOTE_A5 880
#define NOTE_AS5 932 #define NOTE_B5 988 #define NOTE_C6 1047
#define NOTE_CS6 1109 #define NOTE_D6 1175 #define NOTE_DS6 1245
#define NOTE_E6 1319 #define NOTE_F6 1397 #define NOTE_FS6 1480
#define NOTE_G6 1568 #define NOTE_GS6 1661 #define NOTE_A6 1760
#define NOTE_AS6 1865 #define NOTE_B6 1976 #define NOTE_C7 2093
#define NOTE_CS7 2217 #define NOTE_D7 2349 #define NOTE_DS7 2489
#define NOTE_E7 2637 #define NOTE_F7 2794 #define NOTE_FS7 2960
#define NOTE_G7 3136 #define NOTE_GS7 3322 #define NOTE_A7 3520
#define NOTE_AS7 3729 #define NOTE_B7 3951 #define NOTE_C8 4186
#define NOTE_CS8 4435 #define NOTE_D8 4699 #define NOTE_DS8 4978
const int pinTone = 11; //buzzerpin
t
    int note;
    int tempo;
} notesType;

```




```

int pinControlIn = 12;
int controlDelay = 1000;
unsigned long controlPrevTime = 0;
int _playNote = 0;
#define LATCH 9
#define CLOCK 10
#define DATA 8
static int led = 0;
byte number[23] = {0b00000000,
                  0b00000001,
                  0b00000011,
                  0b00000111,
                  0b00001110,
                  0b00011100,
                  0b00111000,
                  0b01110000,
                  0b11100000,
                  0b11000000,
                  0b10000000,
                  0b00000000,
                  0b10000000,
                  0b11000000,
                  0b11100000,
                  0b01110000,
                  0b00111000,
                  0b00011100,
                  0b00001110,
                  0b00000111,
                  0b00000011,
                  0b00000001,
                  0b00000000
                };
void setup()
{
  pinMode(pinTone, OUTPUT);
  totalNotes = sizeof(aNotes) / sizeof(notesType); pinMode(LATCH, OUTPUT);
  pinMode(CLOCK, OUTPUT);
  pinMode(DATA, OUTPUT);
  pinMode(LATCH, OUTPUT);
}
void loop()
{
  mission1();
  mission2();
}
void mission1() {
  static unsigned long time = millis();
  if (millis() - time >= 80 && led <= 22) {
    time = millis();
    led++;
    digitalWrite(LATCH, LOW);
    shiftOut(DATA, CLOCK, MSBFIRST, number[led]);
    digitalWrite(LATCH, HIGH);
  }
  if (led == 22) {
    led = 0;
  }
}
void mission2() {
  curTime = millis();
  if (curTime - controlPrevTime > controlDelay) {
    controlPrevTime = curTime;
    digitalWrite(pinControlOut, LOW);
    _playNote = 1;
  }
  if (curTime - tonePrevTime >= noteDelay && _playNote == 1) {
    noteIndex++;
    tonePrevTime = curTime;
    noTone(pinTone);
    tone(pinTone, aNotes[noteIndex].note, aNotes[noteIndex].tempo);
    noteDelay = aNotes[noteIndex].tempo;
    if (noteIndex >= totalNotes) {
      noteIndex = -1;
      noteDelay = 10;
    }
  }
}
void resetTones() {
  _playNote = 0;
  noteIndex = -1;
  noteDelay = aNotes[(noteIndex + 1)].tempo;
}

```