

This ULU

The *ULU.43 – Ring counter* can be used as an orchestrator. When connected to a pulse generator, for instance the *ULU.11 Clock generator*, it will switch on the eight outputs sequentially and one at the time. The number of active outputs is determined by the value on the input data-bus.

Used parts

Only standard parts are used:

1x casing 80 x 50 x 20mm;
10x 2mm signal connector;
10x black O-ring 9 x 5 x 2mm;
1x 4-bit data connector;
1x colored O-ring 8 x 5 x 1.5mm;
1x power connector;
5x 10K pull-up resistor;

8x 3mm round LED ;
8x resistor to dim the LED;
8x LED holder;
4x micro (G6K-2F-Y-5VDC) relay;
4x fly back diode (1N4148);
1x Arduino Nano;

Construction

The standard ULU specifications are applicable as specified in the datasheet *ULU.00 – Common specifications*.

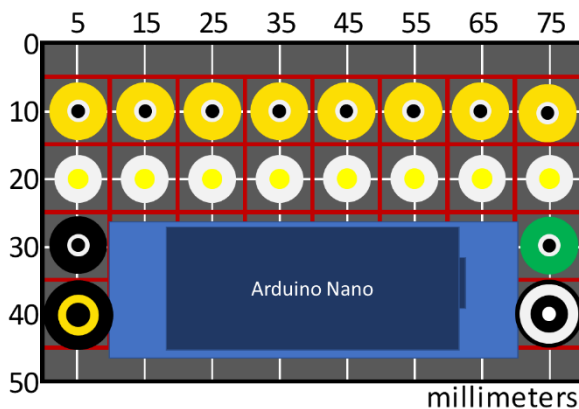


Figure 1 – Drill guide

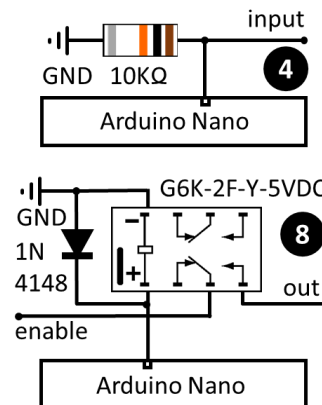


Figure 2 – Used arduino interfaces

Construction of this high-density ULU is straight forward. As shown in Figure 4, the ground of the eight LEDs is interconnected with a thin copper wire. For the relay PCB and the Arduino the familiar book-fold technique is used to make all sides well accesable.

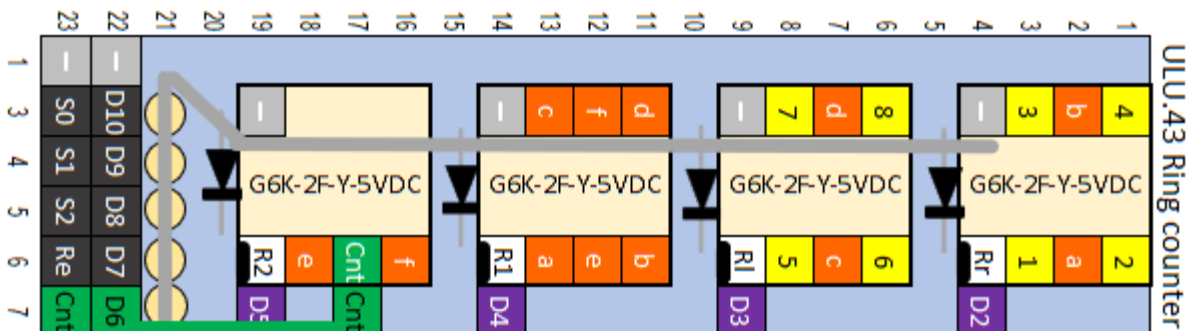


Figure 3 – layout relay PCB

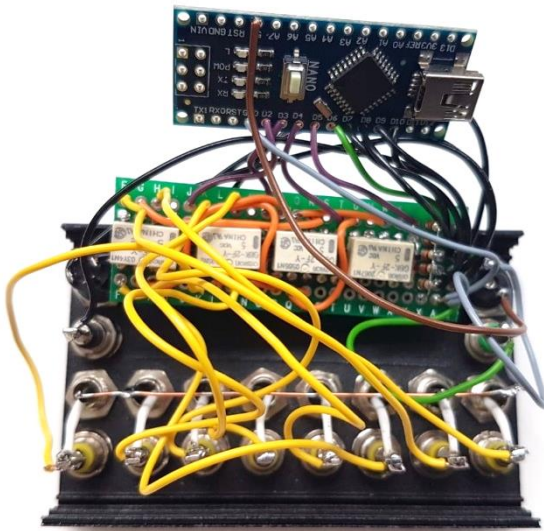


Figure 4 – ULU inside

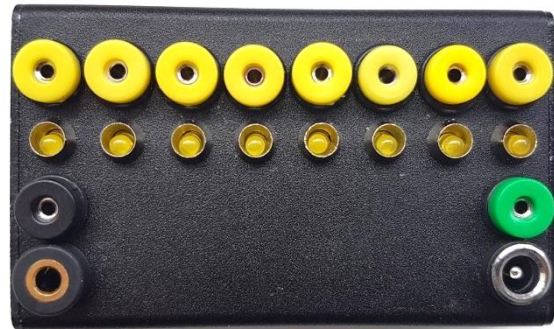


Figure 5 – Finished ULU

	Port	Con.	Rest.	Func.	Interface	Signal
1.	D2	□		O	⑧	Relay r
2.	D3	□		O	⑧	Relay l
3.	D4	□		O	⑧	Relay 1
4.	D5	□		O	⑧	Relay 2
5.	D6	●		I	④	Count
6.	D7	●		I	④	Reset
7.	D8	④		I	④	Set 2
8.	D9	④		I	④	Set 1
9.	D10	④		I	④	Set 0
10.	D13			L		Heartbeat
11.	+5V	⊙	I	I		+5V
12.	GND	⊙	I	I		GND

Input, Output, Led, SPI, Toggle switch, Rotary switch

Figure 6 – Pinout Arduino Nano

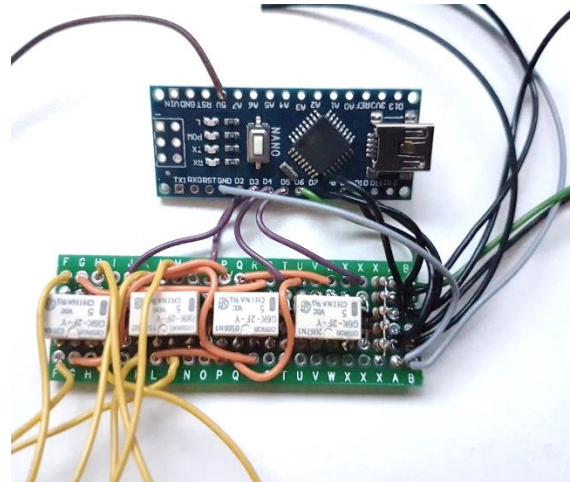


Figure 7 – The PCB

Usage

This ULU can be used as an orchestrator of events that needs to be done in serie. The input data-bus is used to set the maximum value. If no input is provided, the ring counter will use eight outputs. The reset socket can be used to reset the counter back to the first output.

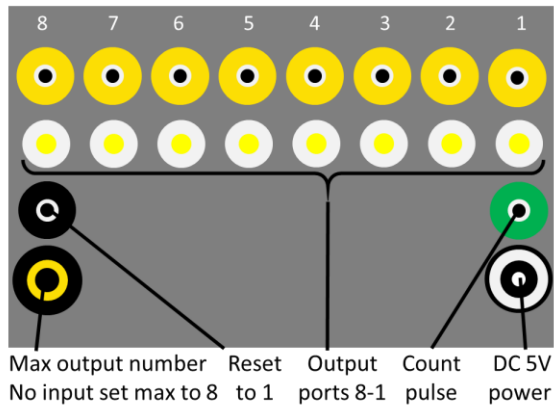


Figure 8 – Controls and connectors

Arduino Nano Solder check

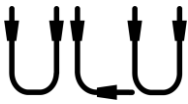
```
/* ULU.43 Ring counter - solder check */
/* CC BY-NC-SA Jeroen Brinkman */

int i, value; // integers

#define BOUNCE 8
#define RELAYR 2
#define RELAYL 3
#define RELAY1 4
#define RELAY2 5
#define COUNT 6
#define RESET 7
#define MAX2 8
#define MAX1 9
#define MAX0 10
#define HEART 13

void setup() {
  pinMode(RELAYR, OUTPUT);
  pinMode(RELAYL, OUTPUT);
  pinMode(RELAY1, OUTPUT);
  pinMode(RELAY2, OUTPUT);
  pinMode(COUNT, INPUT);
  pinMode(RESET, INPUT);
  pinMode(MAX2, INPUT);
  pinMode(MAX1, INPUT);
  pinMode(MAX0, INPUT);
  pinMode(HEART, OUTPUT);
  /* Every ULU with an Arduino introduces itself. This one is speechless, so stays quiet */

  /* Output test */
  for (int i = 1; i < 9; i++) {
    switch (i) {
      case 1: // 000
        digitalWrite(RELAYL, LOW);
        digitalWrite(RELAYR, LOW);
        digitalWrite(RELAY1, LOW);
        digitalWrite(RELAY2, LOW);
        break;
      case 2: // 001
        digitalWrite(RELAYL, HIGH);
        digitalWrite(RELAYR, HIGH);
        digitalWrite(RELAY1, LOW);
        digitalWrite(RELAY2, LOW);
        break;
      case 3: // 010
        digitalWrite(RELAYL, LOW);
        digitalWrite(RELAYR, LOW);
        digitalWrite(RELAY1, HIGH);
        digitalWrite(RELAY2, LOW);
        break;
      case 4: // 011
        digitalWrite(RELAYL, HIGH);
        digitalWrite(RELAYR, HIGH);
        digitalWrite(RELAY1, HIGH);
        digitalWrite(RELAY2, LOW);
        break;
      case 5: // 100
        digitalWrite(RELAYL, LOW);
        digitalWrite(RELAYR, LOW);
        digitalWrite(RELAY1, LOW);
        digitalWrite(RELAY2, HIGH);
        break;
      case 6: // 101
        digitalWrite(RELAYL, HIGH);
        digitalWrite(RELAYR, HIGH);
        digitalWrite(RELAY1, LOW);
        digitalWrite(RELAY2, HIGH);
        break;
      case 7: // 110
        digitalWrite(RELAYL, LOW);
```



```
        digitalWrite(RELAYR, LOW);
        digitalWrite(RELAY1, HIGH);
        digitalWrite(RELAY2, HIGH);
        break;
    case 8: // 111
        digitalWrite(RELAYL, HIGH);
        digitalWrite(RELAYR, HIGH);
        digitalWrite(RELAY1, HIGH);
        digitalWrite(RELAY2, HIGH);
        break;
    }
    delay(500);
}
};

void loop(){
    digitalWrite(HEART, (millis() / 1000) % 2); //1s heartbeat for the onboard led

    /* Read all input */
    value = 1;
    if(digitalRead(MAX0) == HIGH) value = 2;
    if(digitalRead(MAX1) == HIGH) value = 3;
    if(digitalRead(MAX2) == HIGH) value = 4;
    if(digitalRead(RESET) == HIGH) value = 5;
    /* Write output */
    switch (value) {
        case 1: // 000
            digitalWrite(RELAYL, LOW);
            digitalWrite(RELAYR, LOW);
            digitalWrite(RELAY1, LOW);
            digitalWrite(RELAY2, LOW);
            break;
        case 2: // 001
            digitalWrite(RELAYL, HIGH);
            digitalWrite(RELAYR, HIGH);
            digitalWrite(RELAY1, LOW);
            digitalWrite(RELAY2, LOW);
            break;
        case 3: // 010
            digitalWrite(RELAYL, LOW);
            digitalWrite(RELAYR, LOW);
            digitalWrite(RELAY1, HIGH);
            digitalWrite(RELAY2, LOW);
            break;
        case 4: // 011
            digitalWrite(RELAYL, HIGH);
            digitalWrite(RELAYR, HIGH);
            digitalWrite(RELAY1, HIGH);
            digitalWrite(RELAY2, LOW);
            break;
        case 5: // 100
            digitalWrite(RELAYL, LOW);
            digitalWrite(RELAYR, LOW);
            digitalWrite(RELAY1, LOW);
            digitalWrite(RELAY2, HIGH);
            break;
    }
}
```

Arduino Nano program

```
/* ULU.43 Ring counter - program code */
/* CC BY-NC-SA Jeroen Brinkman */

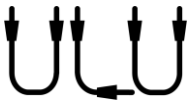
int i, signa, value; // integers
int maximum; // 3-bit words
bool reset, count, lastcount; // booleans

#define BOUNCE 8
#define RELAYR 2
#define RELAYL 3
#define RELAY1 4
#define RELAY2 5
#define COUNT 6
#define RESET 7
#define MAX2 8
#define MAX1 9
#define MAX0 10
#define HEART 13

void setup() {
    pinMode(RELAYR, OUTPUT);
    pinMode(RELAYL, OUTPUT);
    pinMode(RELAY1, OUTPUT);
    pinMode(RELAY2, OUTPUT);
    pinMode(COUNT, INPUT);
    pinMode(RESET, INPUT);
    pinMode(MAX2, INPUT);
    pinMode(MAX1, INPUT);
    pinMode(MAX0, INPUT);
    pinMode(HEART, OUTPUT);
    /* Every ULU with an Arduino introduces itself. This one is speechless, so stays quiet*/

    lastcount = false; value = 1;
};

void loop(){
```



```
digitalWrite(HEART, (millis() / 1000) % 2); //1s heartbeat for the onboard led

/* Read all input */
maximum = digitalRead(MAX0) + (digitalRead(MAX1) * 2) + (digitalRead(MAX2) * 4);
if (maximum == 0) maximum = 9;
reset = (digitalRead(RESET) == HIGH);
signa = 0;
for (int i = 0; i < BOUNCE; i++) {signa += digitalRead(COUNT); delay(2);} // eliminate bouncing
count = (signa > (i / 2));

/* Process input */
if (!count && lastcount) value += 1;
if (value == 9) value = 1;
if (value > maximum) value = 1;
if (reset) value = 1;

/* Write output */
switch (value) {
  case 1: // 000
    digitalWrite(RELAYL, LOW);
    digitalWrite(RELAYR, LOW);
    digitalWrite(RELAY1, LOW);
    digitalWrite(RELAY2, LOW);
    break;
  case 2: // 001
    digitalWrite(RELAYL, HIGH);
    digitalWrite(RELAYR, HIGH);
    digitalWrite(RELAY1, LOW);
    digitalWrite(RELAY2, LOW);
    break;
  case 3: // 010
    digitalWrite(RELAYL, LOW);
    digitalWrite(RELAYR, LOW);
    digitalWrite(RELAY1, HIGH);
    digitalWrite(RELAY2, LOW);
    break;
  case 4: // 011
    digitalWrite(RELAYL, HIGH);
    digitalWrite(RELAYR, HIGH);
    digitalWrite(RELAY1, HIGH);
    digitalWrite(RELAY2, LOW);
    break;
  case 5: // 100
    digitalWrite(RELAYL, LOW);
    digitalWrite(RELAYR, LOW);
    digitalWrite(RELAY1, LOW);
    digitalWrite(RELAY2, HIGH);
    break;
  case 6: // 101
    digitalWrite(RELAYL, HIGH);
    digitalWrite(RELAYR, HIGH);
    digitalWrite(RELAY1, LOW);
    digitalWrite(RELAY2, HIGH);
    break;
  case 7: // 110
    digitalWrite(RELAYL, LOW);
    digitalWrite(RELAYR, LOW);
    digitalWrite(RELAY1, HIGH);
    digitalWrite(RELAY2, HIGH);
    break;
  case 8: // 111
    digitalWrite(RELAYL, HIGH);
    digitalWrite(RELAYR, HIGH);
    digitalWrite(RELAY1, HIGH);
    digitalWrite(RELAY2, HIGH);
    break;
}
lastcount = count;
}
```