

PLAN DE ACTIVIDADES DEL CURSO DE ELECTRÓNICA INDUSTRIAL APLICADA

Prof. Oscar NÚÑEZ Mori

INTRODUCCIÓN

En el presente curso se implementarán tres (3) proyectos electrónicos básicos, uno por unidad didáctica donde los alumnos pondrán en práctica los conocimientos teóricos adquiridos en el curso e implementados a nivel virtual en los simuladores de código abierto, llevándolos a la realidad. Para ello los alumnos usarán componentes y equipos electrónicos de fácil adquisición y bajo costo. De esta manera los estudiantes adquirirán el dominio del manejo práctico del hardware que utilizarán en circuitos y sistema eléctricos y electrónicos en su ámbito profesional.

OBJETIVOS

Unidad I: Proyecto de un Circuito Electrónico usando Componentes de Electrónica de Potencia

Unidad II: Proyecto de un Sistema Electrónico usando Microcontroladores.

Unidad III: Proyecto de un Sistema Electrónico usando Tecnología PWM.

DESARROLLO

A continuación hacemos el listado de equipos básicos, herramientas y materiales de bajo costo a ser utilizados en todos los proyectos. Luego veremos en detalle cada proyecto con su respectiva lista de componentes electrónicos.

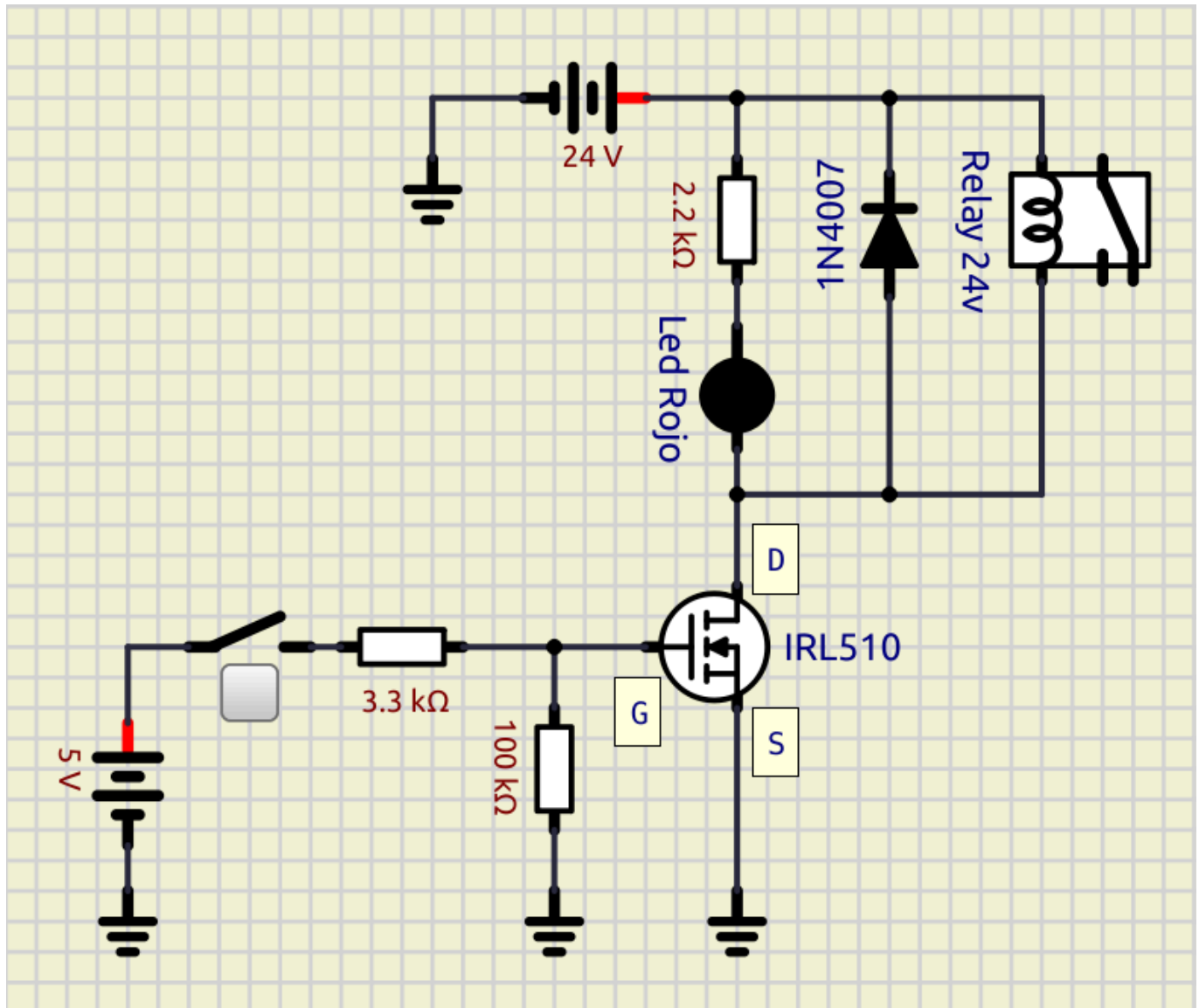
EQUIPOS, HERRAMIENTAS Y MATERIALES

01. Multímetro Digital (Con Voltímetro, Amperímetro, Ohmímetro, Medidor de Semiconductores.)
02. Cautín de 30 Watts para uso electrónico.
03. Soporte para soldar con lupa y luz.
04. Desoldador.
05. Alicates tipo pinza para uso electrónico.
06. Alicates de Corte para uso electrónico.
07. Pelacables.
08. Protoboard. (2 unidades)
09. Cinta Aislante.
10. Cables sólidos de colores para uso electrónico.
11. Soldadura de Estaño con núcleo de Resina 60/40 para uso electrónico.
12. Tabla de madera tamaño A4.
13. Batería de 12 Voltios. (2 Unidades)
14. Batería de 1.5 Voltios Tipo D. (4 unidades)
15. Cable con un cocodrilo en cada punta (6 unidades; rojo, negro, blanco, Etc)
16. Placa de baquelita perforada.
17. Extensión eléctrica con Adaptador (para UNJ)
18. Cargador de Batería de 12 Voltios.
19. Alcohol isopropílico preferentemente o Alcohol del más alto grado 95 o 97 grados.
20. Estilete o Cutter
21. Borrador
22. Pistola con cartucho de silicona para bricolaje.
23. Limpiador de puntas para cautín (hilos metálicos)

CIRCUITOS PROPEDEUTICOS

El circuito de mando lo representa la malla de 5 Volt., que puede perfectamente ser excitado con 4 pilas de 1.5 volt y utiliza un Mosfet IRL510, con un VGS de hasta 10 V. los cuales no se debe sobrepasar pues se quemaría el MOSFET. Por lo que excitarlo con 5 a 7 voltios estará bien. El circuito de fuerza está representado por el Relay de 24 V. a 220 V. pero puede ser perfectamente un Contactor con características similares.

Circuito Switch con Mosfet

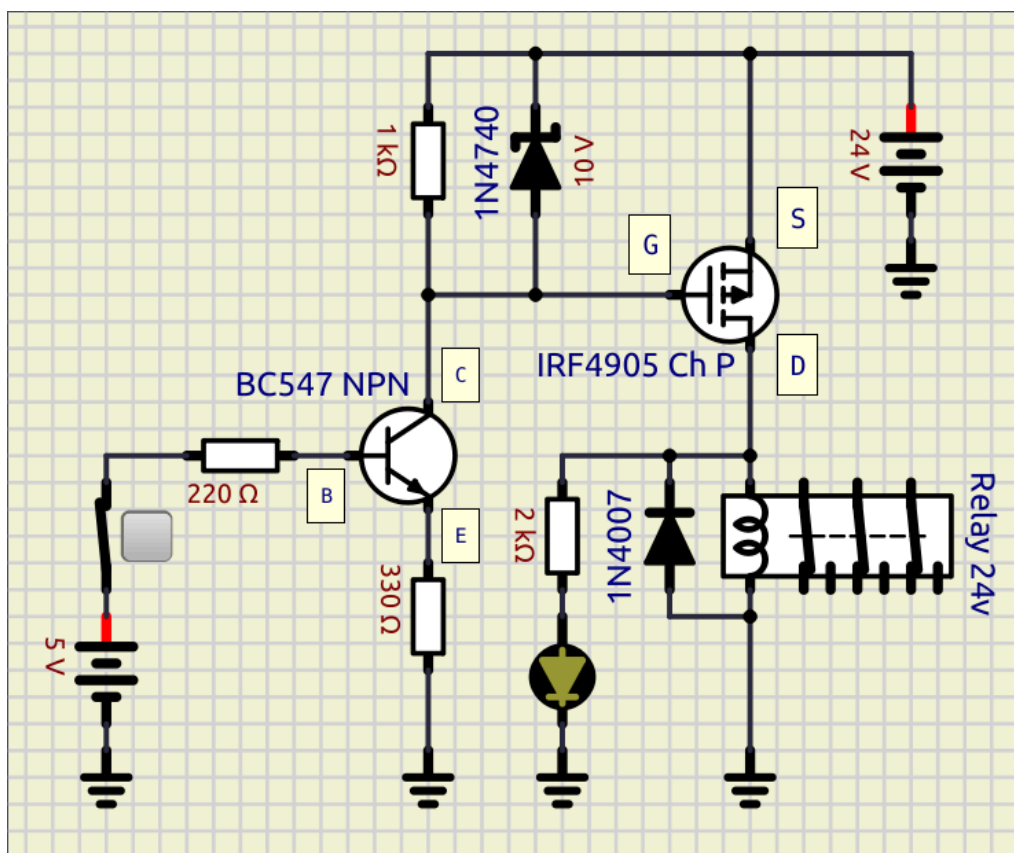


Lista de componentes para la implementación

- 2 Baterías de 12 Voltios
- 4 pilas de 1.5 volt.
- 2 Resistencias de 3.3 Kohm
- 2 Resistencias de 100 Kohm
- 2 Resistencias de 2.2 Kohm
- 2 IRL510 (MOSFET Canal N) o equivalente
- 1 Led Rojo
- 1 diodo rectificador 1N4007
- 1 Relay de 24 volt. .a 220 volt.

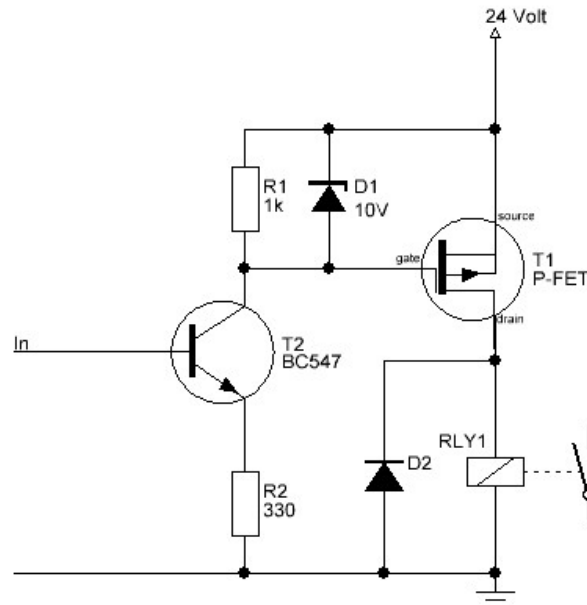
UNIDAD I**1. PROYECTO DE UN CIRCUITO ELECTRÓNICO USANDO COMPONENTES DE ELECTRÓNICA DE POTENCIA****1.1. CONTROL ON/OFF DE MOTOR DC VOLTIOS MEDIANTE MOSFET DE POTENCIA**

En el siguiente circuito basado en (Using FET's to Switch a load, 2015), podemos apreciar el control de encendido y apagado de un contactor trifásico de 24v en su bobina de control y tres polos a 30 amperios

**Lista de Componentes para su implementación**

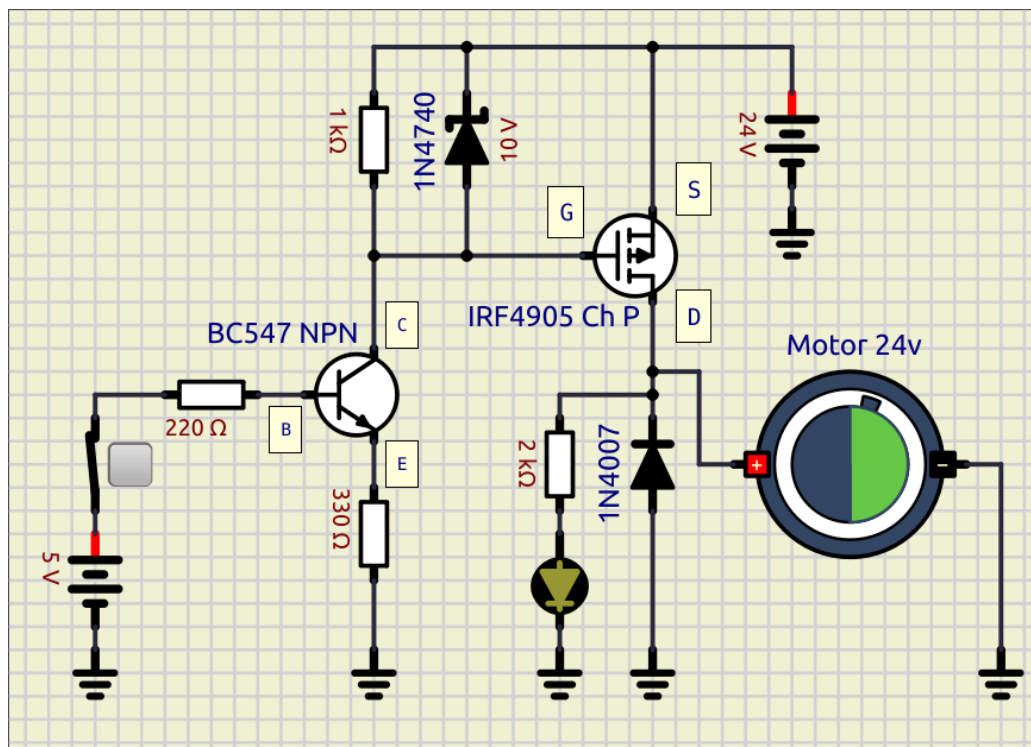
- 1 transistor BC547 NPN
- 1 MOSFET Canal P, IRF4905
- Resistores de 220 ohm, 1 Kohm, 2 Kohm, 330 ohm
- 1 Diodo Zener de 10v., 1N4740
- 1 Diodo rectificador 1N4007
- 1 Diodo LED Rojo
- 1 Contactor 24v a 220v, 30amp, de 3 polos
- 1 switch tipo botón.

El circuito original lo podemos apreciar en la siguiente imagen



(Using FET's to Switch a load, 2015)

Este mismo circuito nos permite también controlar un motor DC de 24 v.

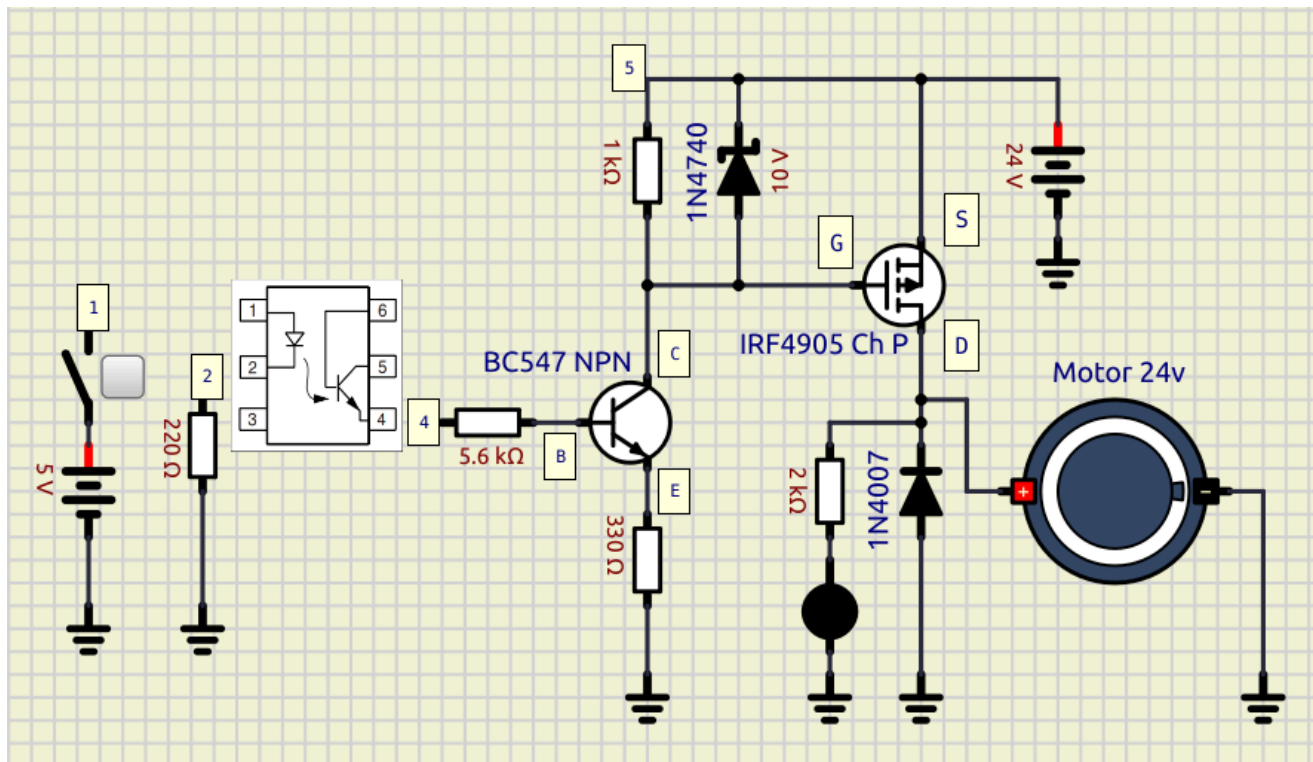


Basado en (Using FET's to Switch a load, 2015)

Lista de Componentes para su implementación

- 1 transistor BC547 NPN
- 1 MOSFET Canal P, IRF4905
- Resistores de 220 ohm, 1 Kohm, 2 Kohm, 330 ohm
- 1 Diodo Zener de 10v., 1N4740
- 1 Diodo rectificador 1N4007
- 1 Diodo LED Rojo
- 1 Motor de 24v.
- 1 switch tipo botón.

Una variante que nos protege la etapa de control ópticamente, en caso que deseemos usar una placa arduino Uno R3, se presenta a continuación.

Circuito con Optoacoplador, BJT y MOSFET

Basado en (Using FET's to Switch a load, 2015)

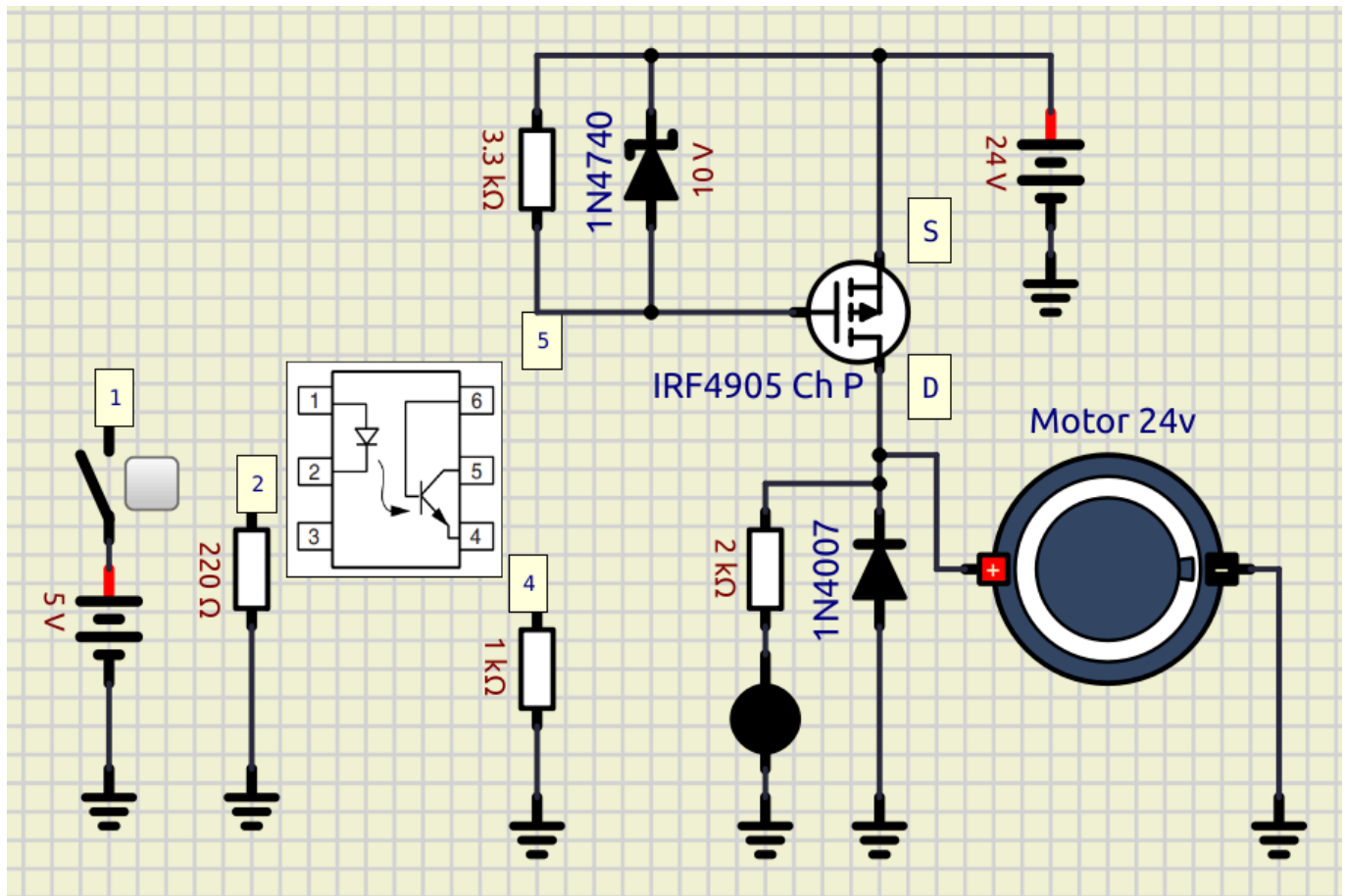
Lista de Componentes para su implementación

- 1 transistor BC547 NPN
- 1 MOSFET Canal P, IRF4905
- Optoacoplador 4N25 o CT817C
- Resistores de 220 ohm, 1 Kohm, 2 Kohm, 330 ohm, 5.6 Kohm
- 1 Diodo Zener de 10v., 1N4740
- 1 Diodo rectificador 1N4007

- 1 Diodo LED Rojo
- 1 Motor de 24v.
- 1 switch tipo botón.

Otra modificación que podríamos hacer se presenta en el siguiente circuito, el cual nos ahorraría usar un Transistor NPN.

Circuito con Optoacoplador, BJT, MOSFET Canal N y Diodo Zener

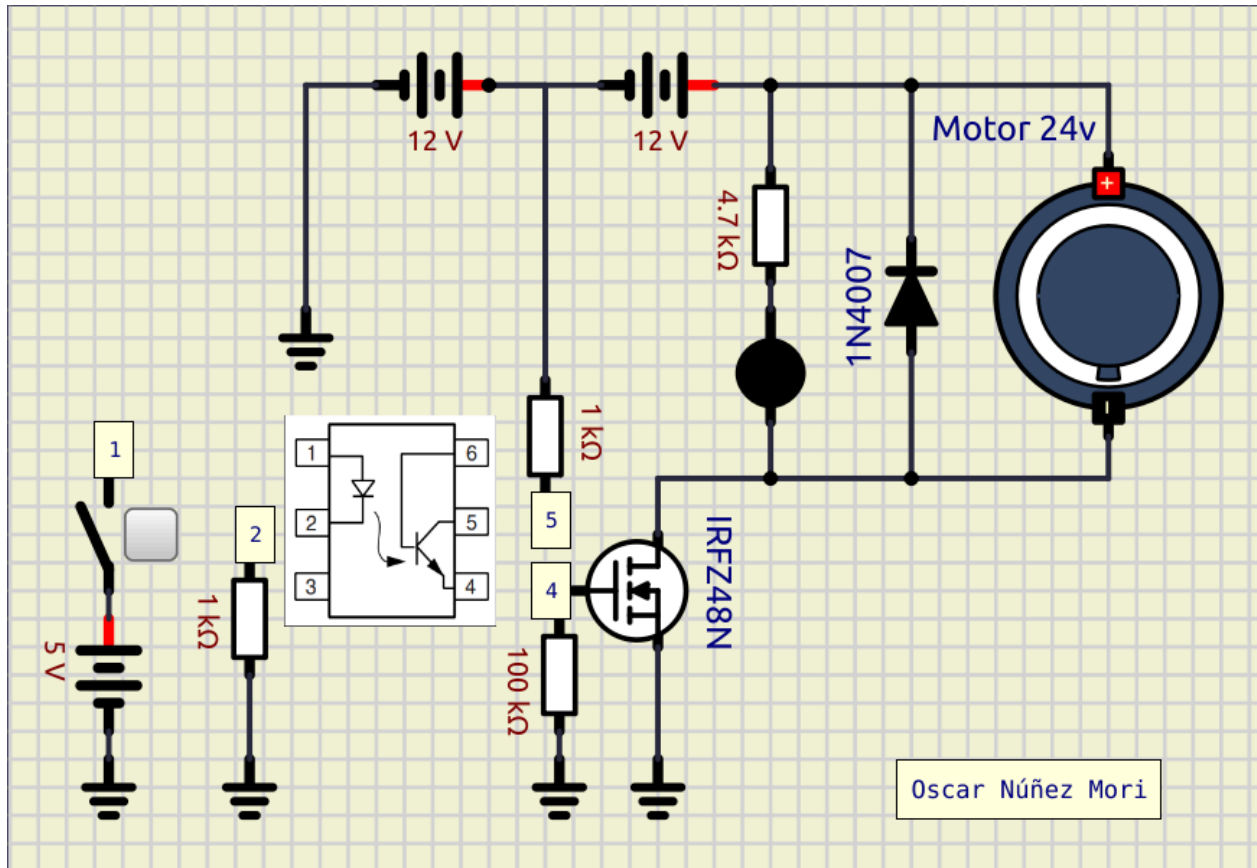


Basado en (Using FET's to Switch a load, 2015)

Lista de Componentes para su implementación

- 1 MOSFET Canal P, IRF4905
- Optoacoplador 4N25 o CT817C
- Resistores de 220 ohm, 1 Kohm, 2 Kohm, - 1 transistor BC547 NPN 3.3 Kohm
- 1 Diodo Zener de 10v., 1N4740
- 1 Diodo rectificador 1N4007
- 1 Diodo LED Rojo
- 1 Motor de 24v.
- 1 switch tipo botón.

Pero si por alguna razón no podemos encontrar en el mercado un MOSFET de Canal N, podemos implementar el siguiente circuito adaptado de (Opto Couplers, s.f.).

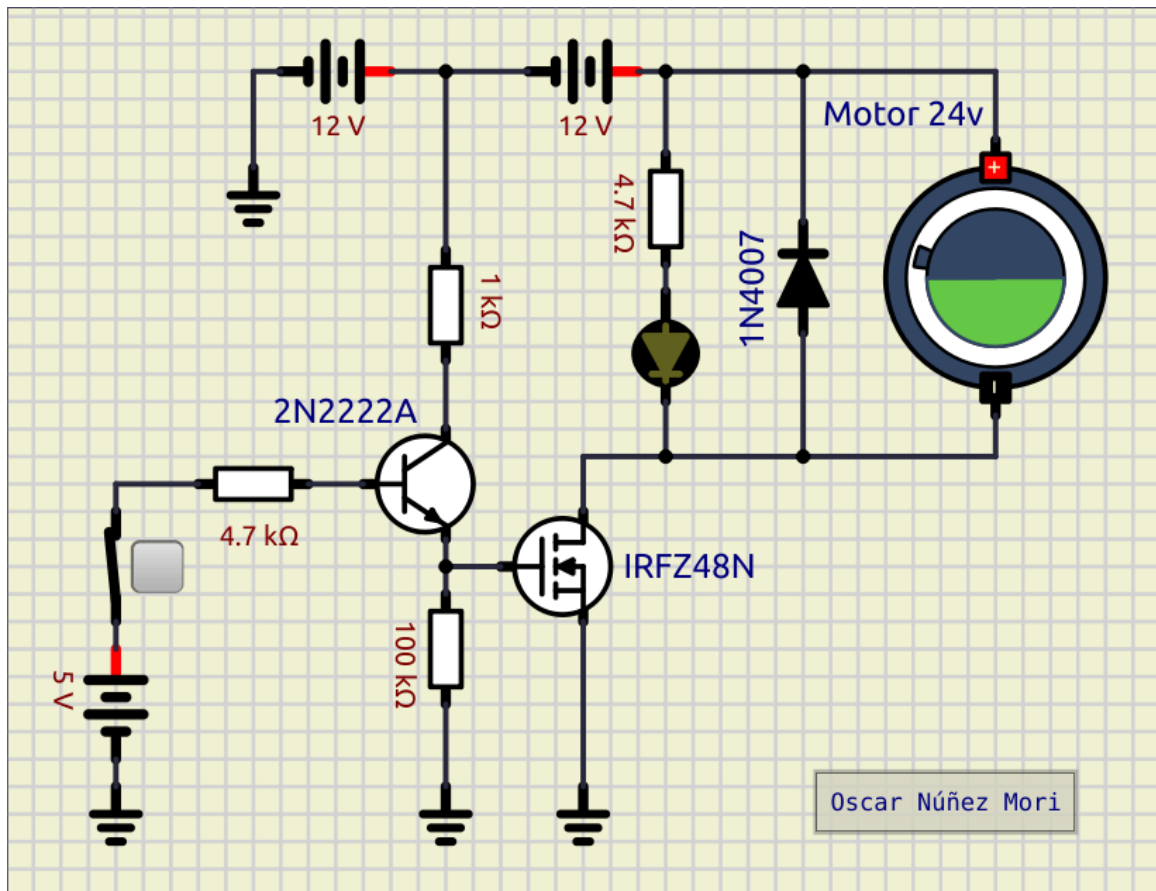


Basado en (Opto Couplers, s.f.)

Lista de Componentes para su implementación

- 1 MOSFET Canal N, IRFZ48N
- Optoacoplador 4N25 o CT817C
- Resistores de 4.7 Kohm, 100 Kohm
- 2 Resistores de 1 Kohm
- 1 Diodo rectificador 1N4007
- 1 Diodo LED Rojo
- 1 Motor de 24v.
- 1 switch tipo botón.

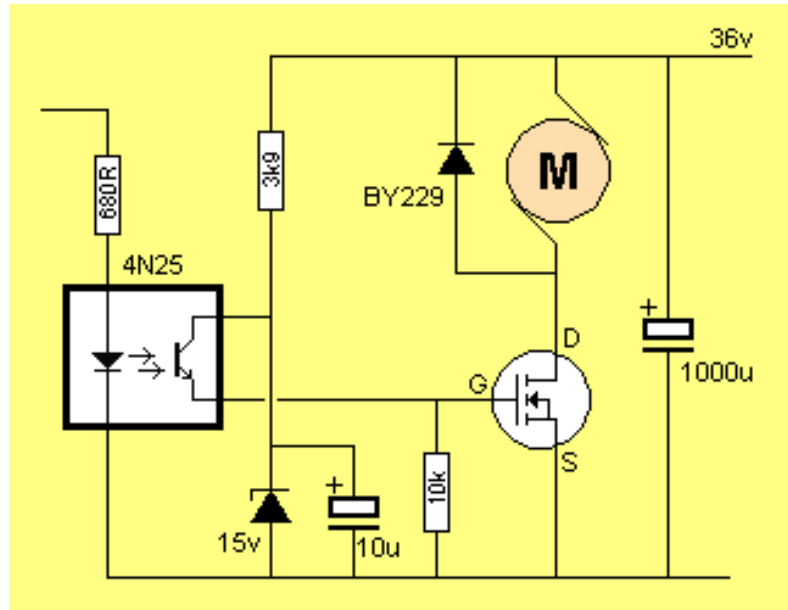
Si por algún motivo no podemos conseguir el Optoacoplador 4N25 o el CTT817C, podemos usar el siguiente circuito:



Lista de Componentes para su implementación

- 1 MOSFET Canal N, IRFZ48N
- 1 Transistor BJT NPN, 2N2222A o equivalente.
- Resistores de 1 Kohm, 100 Kohm
- 2 Resistores de 4.7 Kohm
- 1 Diodo rectificador 1N4007
- 1 Diodo LED Rojo
- 1 Motor de 24v.
- 1 switch tipo botón.

Tomamos la idea de los circuitos anteriores, del siguiente circuito.



(Opto Couplers, s.f.)

LENGUAJE DE PROGRAMACIÓN

El Lenguaje de Programación para codificar la Tarjeta Arduino UNO R3 está fuera del alcance de este curso, pero se ha contemplado material de apoyo que puede ser descargado en los siguientes enlaces:

Núñez, O. (2024a). **Lenguaje INO.**

https://www.academia.edu/114824113/LENGUAJE_INO

Núñez, O. (2024b). **Displays con Arduino UNO R3.**

https://www.academia.edu/114832827/DISPLAYs_con_ARDUINO_UNO_R3

Núñez, O (2024c). **Simulador de Circuitos SIMULIDE.**

https://www.academia.edu/115083010/Simulador_de_Circuitos_SIMULIDE

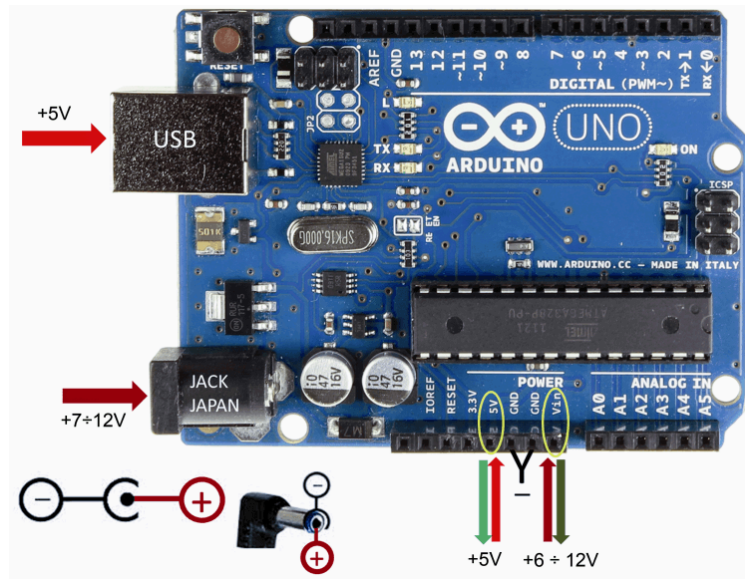
Núñez, O (2024d). **El Simulador WOKWI.**

https://www.academia.edu/115046523/El_Simulador_WOKWI

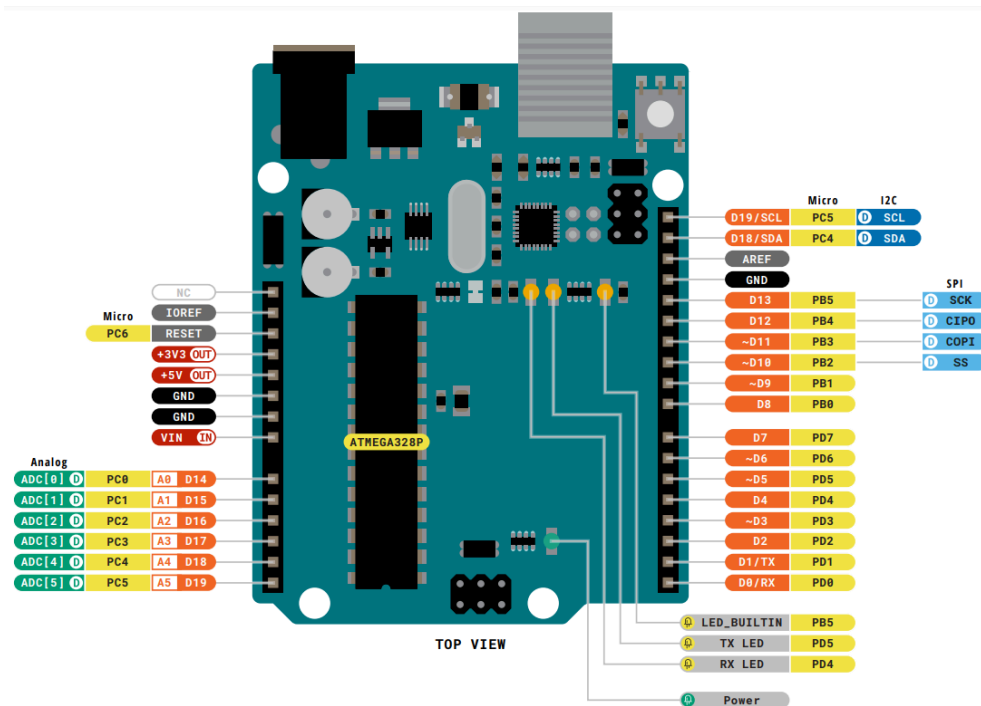
UNIDAD II

2. PROYECTO DE UN SISTEMA ELECTRÓNICO USANDO MICROCONTROLADORES

En el presente proyecto se utilizará la Tarjeta Arduino UNO R3, por su bajo costo y por poseer un Microcontrolador de 8 bits muy potente llamado ATMEGA328P como se aprecia en la siguiente imagen.



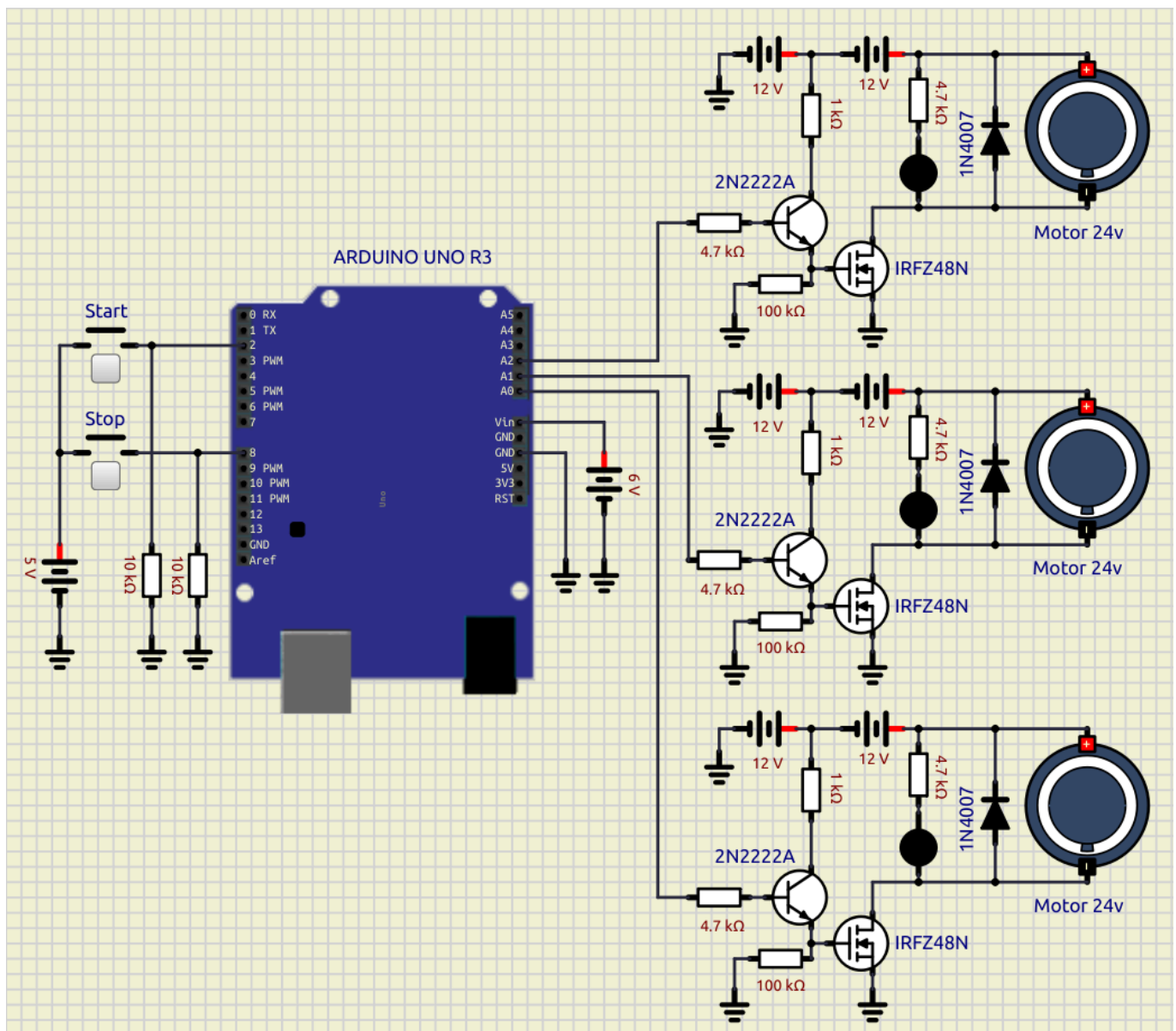
(Geek Factory, 2013-2023)



(Arduino, s.f.)

2.1. ARRANQUE SECUENCIAL DE MOTORES

En el siguiente circuito para la etapa de control en lugar de un PLC se está utilizando la Tarjeta Arduino Uno R3, la cual al ingresar el Código de Programación adecuado en su memoria interna permitirá prender con el botón **Start** los motores en forma secuencial y con el Botón **Stop** los apagará también secuencialmente. Se está trabajando con 24v. pues es un estándar industrial para los circuitos de mando en aplicaciones industriales.





CÓDIGO

ARRANQUE_TRES.ino

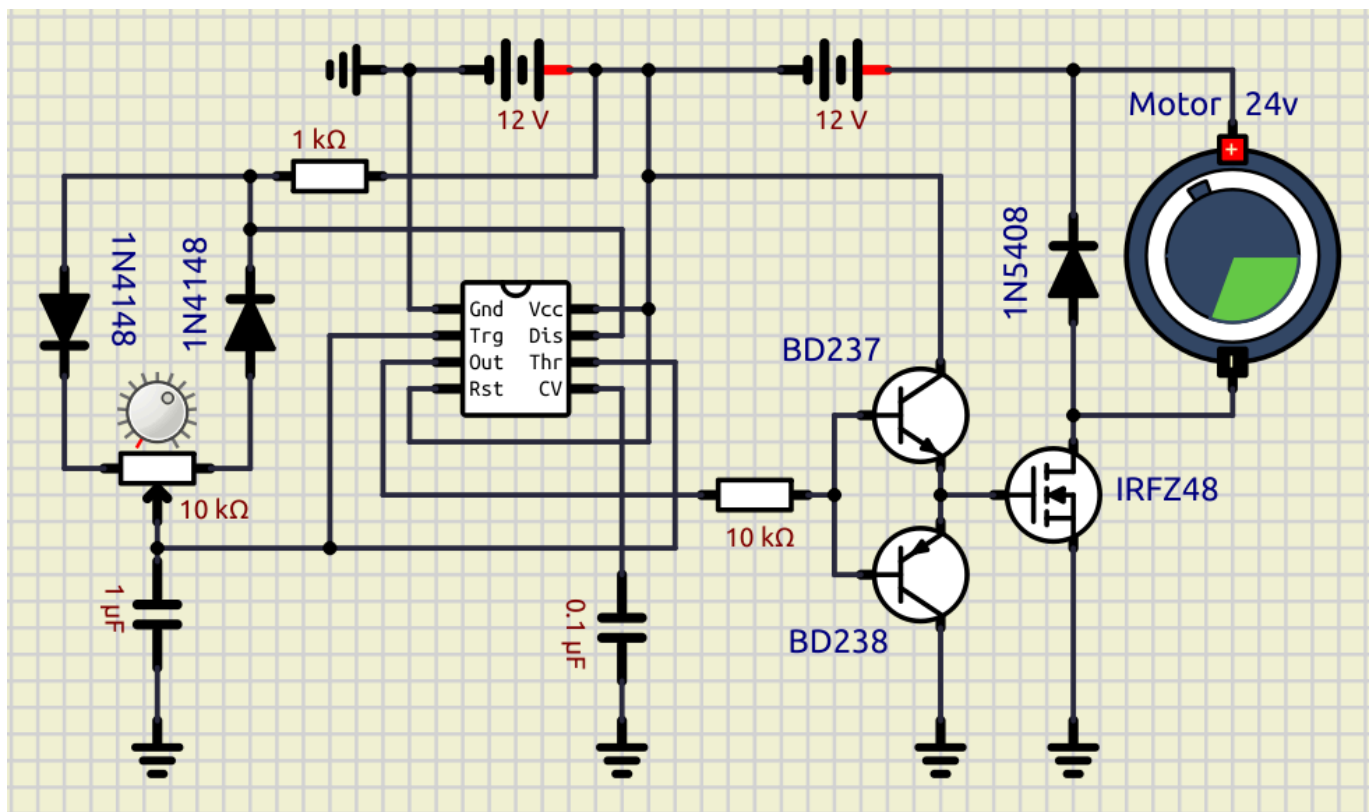
```
1  // Autor. Oscar NÚÑEZ MORI. Jaén, 25-Junio-2024
2  // Definir los pines de los pulsadores y Motores
3  const int Arranque = 2;
4  const int Parada    = 8;
5  const int Motor01   = 14; // A0
6  const int Motor02   = 15; // A1
7  const int Motor03   = 16; // A2
8
9  void setup() {
10
11     // Configurar pines de entrada y salida
12     pinMode(Arranque, INPUT);
13     pinMode(Parada,    INPUT);
14     pinMode(Motor01, OUTPUT);
15     pinMode(Motor02, OUTPUT);
16     pinMode(Motor03, OUTPUT);
17
18 }
19
20 void loop() {
21
22     if( digitalRead(Arranque) ) {    // Prende motores secuencialmente
23
24         digitalWrite(Motor01, HIGH); // Enciende Motor 01
25         delay(3000); // Espera
26
27         digitalWrite(Motor02, HIGH); // Enciende Motor 01
28         delay(3000); // Espera
29
30         digitalWrite(Motor03, HIGH); // Enciende Motor 01
31         delay(3000); // Espera
32     }
33
34     if( digitalRead(Parada) ) {      // Apaga motores secuencialmente
35
36         digitalWrite(Motor01, LOW); // Apaga Motor 01
37         delay(3000); // Espera
38
39         digitalWrite(Motor02, LOW); // Apaga Motor 01
40         delay(2000); // Espera
41
42         digitalWrite(Motor03, LOW); // Apaga Motor 01
43         delay(3000); // Espera
44
45     }
46 }
47
```


Lista de Componentes para su implementación

- 2 Baterías de 12V.
- 6 pilas 1.5v.
- 1 Tarjeta o Placa Arduino UNO R3
- 3 MOSFETs Canal N, IRFZ48N o Equivalente
- 3 Transistor BJT NPN, 2N2222A o equivalente.
- 2 Resistores de 10 Kohm
- 3 Resistores de 1 Kohm
- 3 Resistores de 100 Kohm
- 6 Resistores de 4.7 Kohm
- 3 Diodo rectificador 1N4007 (1 Amperio) o 1N5408 (3 amperios)
- 3 Diodos LED Rojo
- 3 Motor de 24v.
- 3 Relés o Contactores de 24 Voltios
- 2 Pulsadores.

UNIDAD III**3. PROYECTO DE UN SISTEMA ELECTRÓNICO USANDO TECNOLOGÍA PWM****3.1. CIRCUITO ELECTRÓNICO CON PWM**

El presente circuito permite controlar mediante un CI. 555 la Velocidad de un Motor de 24 voltios usando Pulsos Controlados (PWM) en su pin OUT mediante un Potenciómetro. El presente circuito permite controlar mediante un CI. 555 la Velocidad de un Motor de 24 voltios usando Pulsos Controlados (PWM) en su pin OUT mediante un Potenciómetro.



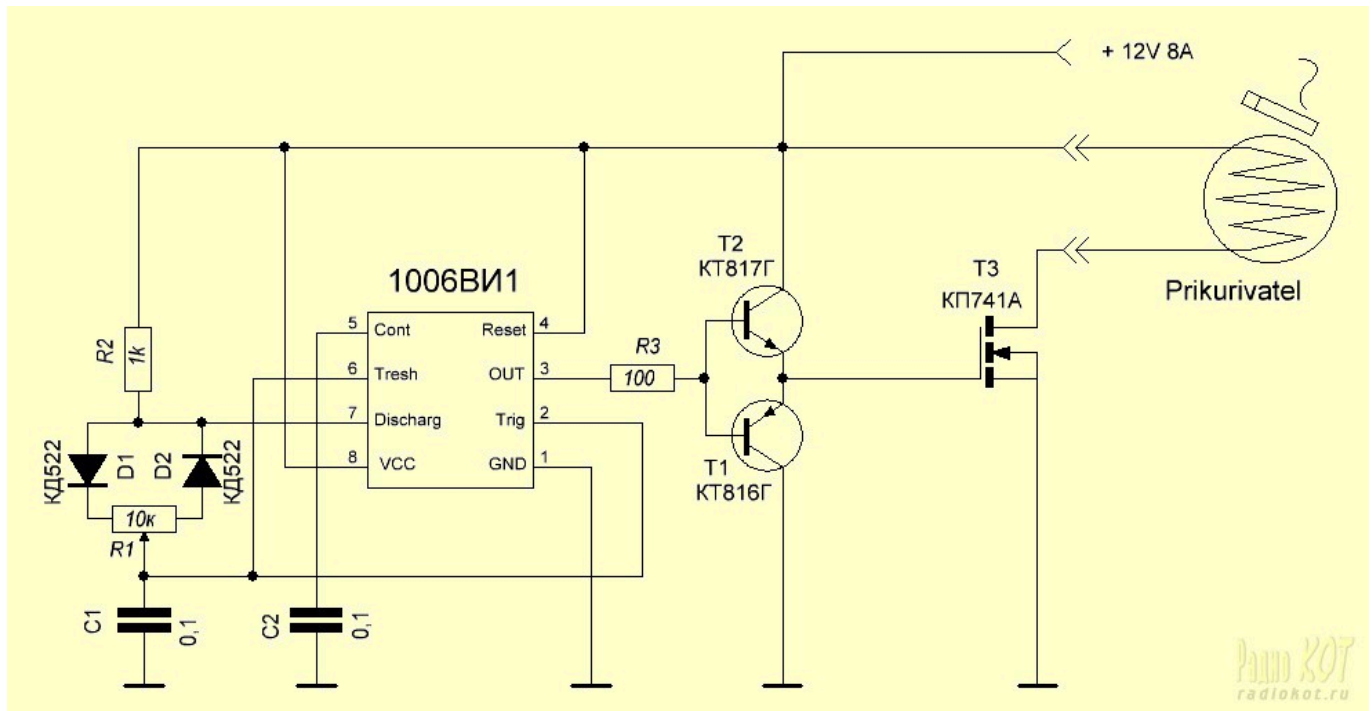
Basado en Поделкин Самоделкин (2016)

Lista de Componentes para su implementación

- 2 Baterías de 12V.
- 1 CI NE555
- 1 Transistor BJT NPN, BD237
- 1 Transistor BJT PNP, BD238
- 1 Transistor BJT NPN BC547
- 1 Transistor BJT PNP BC557
- 1 MOSFET Canal N, IRFZ48 o equivalente
- 2 Diodos de alta velocidad 1N4148
- 1 diodo rectificador 1N4007 (1 Amperio) o 1N5408 (3 Amperios)
- 2 Resistores de 10 Kohm

- 2 Resistores 1 Kohm
- 2 Condensadores electrolíticos 1uf 25v.
- 2 Condensadores Cerámicos de 0.1 uF.
- 1 Potenciometro de 10 Khm
- 1 Potenciometro de 50 Khm

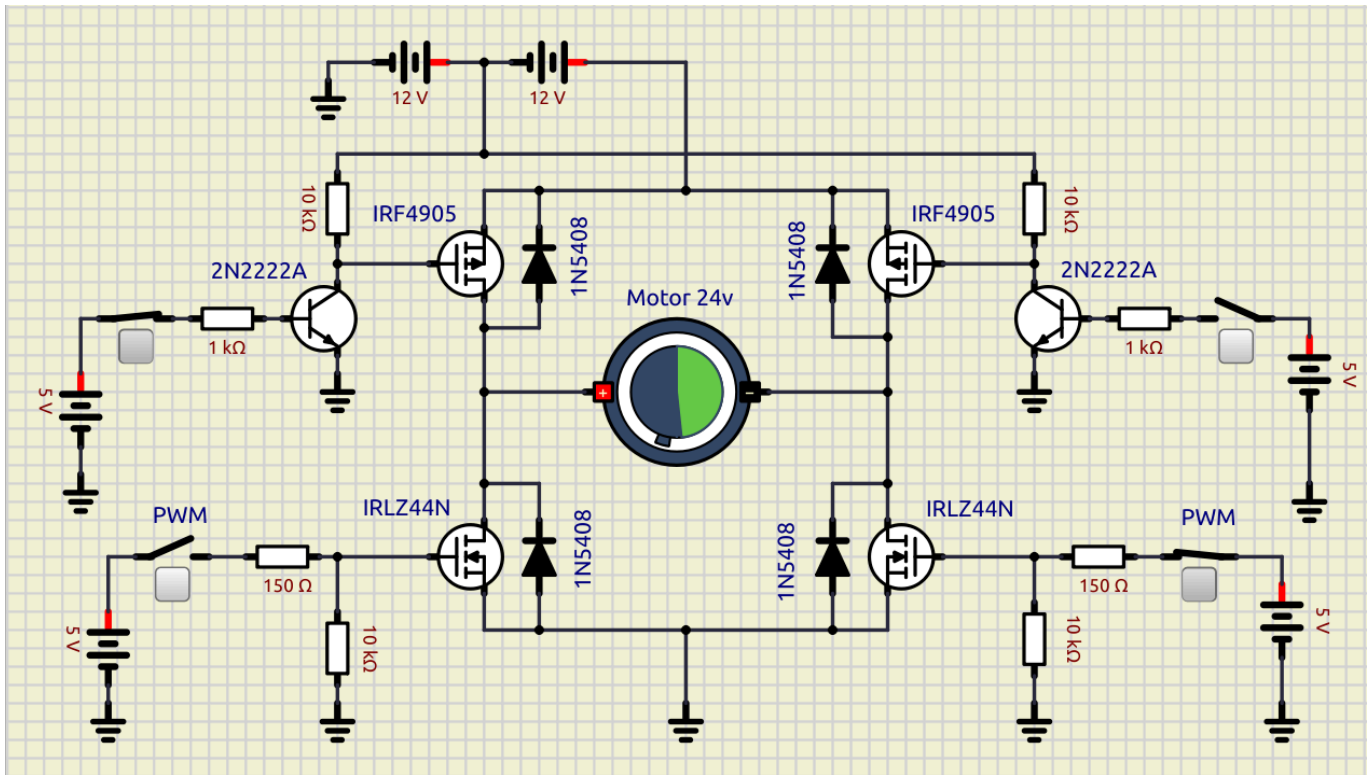
El circuito anterior se basó en el siguiente circuito.



ШИМ на ne 555 (Поделкин Самоделкин, 2016)

Si queremos que el circuito pueda invertir el giro del motor y controlar también su velocidad mediante la técnica de PWM se presenta el siguiente circuito.

Control PWM de Motor DC con Inversión de Giro.

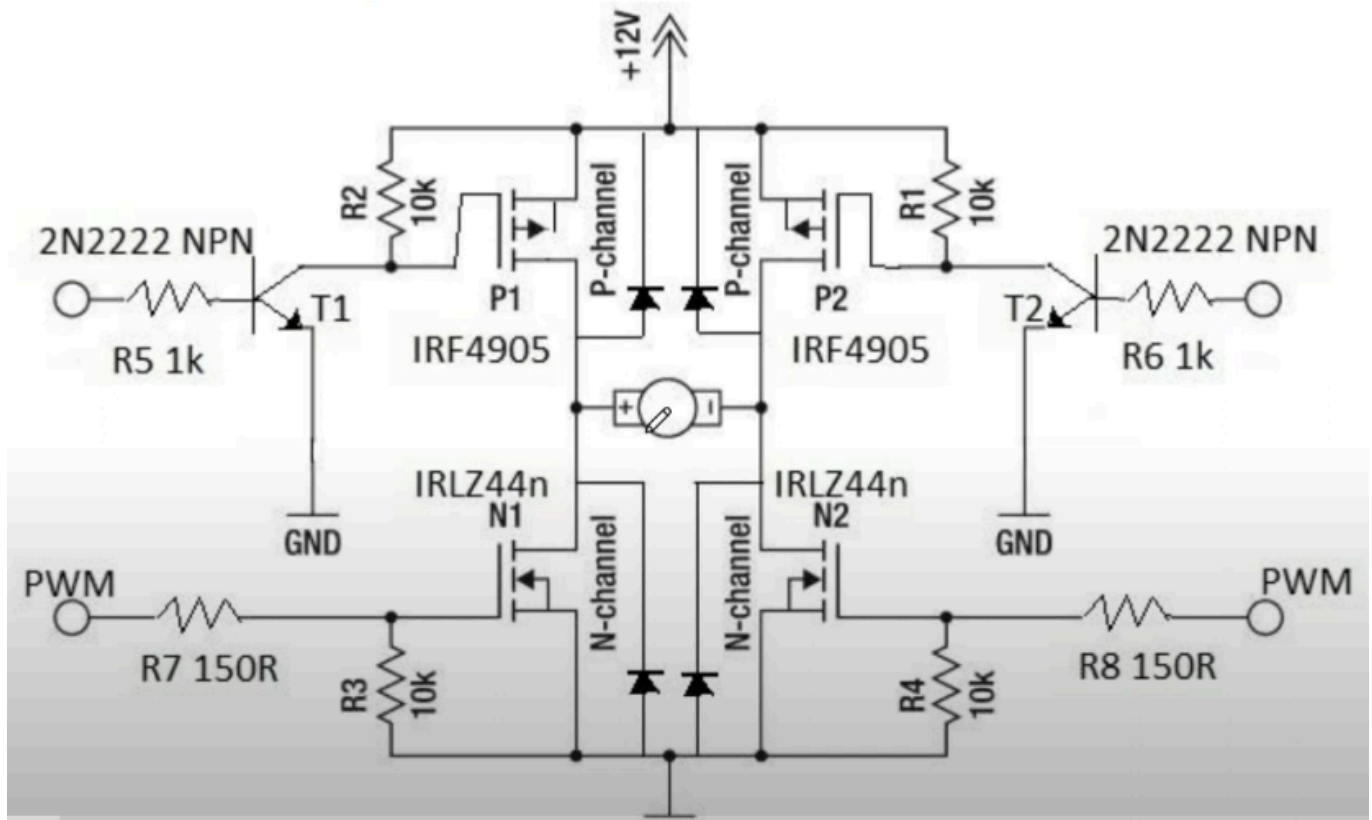


Basado en (Portal Mecatrónica, 2020)

Lista de Componentes para su implementación

- 2 Baterías de 12V.
- 6 pilas de 1.5v.
- 4 Transistores BJT NPN, 2N2222A
- 2 MOSFETs Canal P, IRF4905
- 2 MOSFETS Canal N, IRLZ44N o **IRL540**
- 2 Resistores de 150 Ohm
- 4 Resistores de 10 Kohm
- 2 Resistores de 1 Kohm
- 4 Interruptores tipo Pulsador
- 4 diodos Rectificadores 1N4007 (1 Amperio) o 1N5408 (3 Amperios)
- 1 Motor de 24 V.

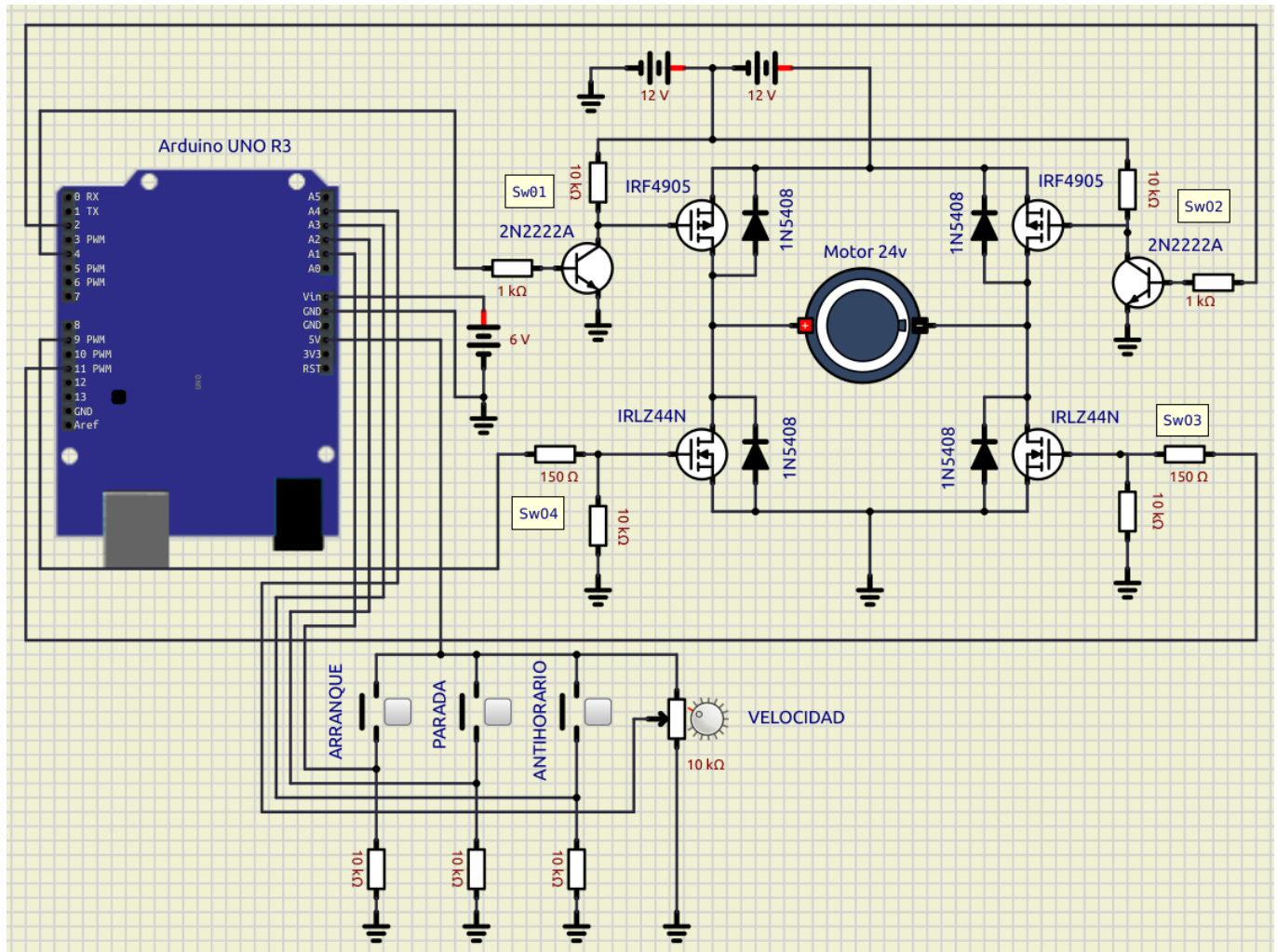
A continuación presentamos el circuito original que se usó en el diseño del circuito anterior



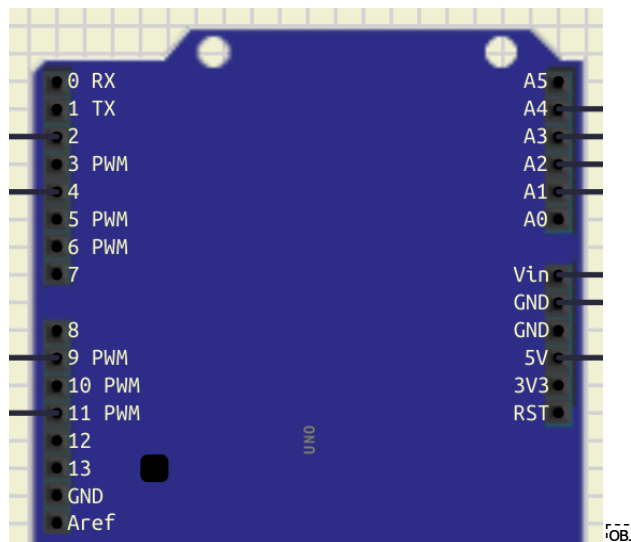
(Portal Mecatrónica, 2020)

Ahora si agregamos una Tarjeta Arduino UNO R3 y lo programamos obtendremos el siguiente circuito:

Control PWM Microprocesado de Motor DC



Basado en (Portal Mecatrónica, 2020)



Lista de Componentes para su implementación

- 2 Baterías de 12 V.
- 6 pilas de 1.5 V.
- 1 Tarjeta Arduino UNO R3
- 4 Transistores BJT NPN, 2N2222A
- 2 MOSFETs Canal P, IRF4905
- 2 MOSFETS Canal N, IRLZ44N o **IRL540**
- 2 Resistores de 150 Ohm
- 7 Resistores de 10 Kohm
- 2 Resistores de 1 Kohm
- 4 diodos Rectificadores 1N4007 (1 Amperio) o 1N5408 (3 Amperios)
- 1 Motor de 24 V.
- 3 pulsadores
- 1 Potenciómetro de 10 Kohm



CÓDIGO

VARIADOR-DE-VELOCIDAD.ino

```
1  // Autor. Oscar NÚÑEZ MORI. Jaén, 25-Junio-2024
2  // Definir los pines de los pulsadores y Motores
3  const int ArranqueHorario      = 15;    // Pin A1  Analog
4  const int Parada                = 16;    // Pin A2  Analog
5  const int ArranqueAntiHorario  = 17;    // Pin A3  Analog
6  const int Velocidad             = 18;    // Pin A4  Analog
7  const int Switch01 = 4;    // Pin 4  Sw01  Digital
8  const int Switch02 = 2;    // Pin 2  Sw02
9  const int Switch03 = 11;   // Pin 11 PWM Sw03
10 const int Switch04 = 9;    // Pin 9  PWM SW04
11 // Variables
12 int Memoria;
13 int Pwm;
```

```
14
15 void setup() {
16
17     // Configurar pines de entrada
18     pinMode(ArranqueHorario, INPUT);
19     pinMode(Parada, INPUT);
20     pinMode(ArranqueAntiHorario, INPUT);
21     pinMode(Velocidad, INPUT);
22     // Configurar pines de salida
23     pinMode(Switch01, OUTPUT);
24     pinMode(Switch02, OUTPUT);
25     pinMode(Switch03, OUTPUT);
26     pinMode(Switch04, OUTPUT);
27
28 }
```

```
29
30 void loop() {
31
32     if( analogRead(ArranqueHorario) ) { //
33         delay(100);
34         digitalWrite(Switch02, LOW); //
35         digitalWrite(Switch04, LOW); //
36         delay(1000); //spera
37         digitalWrite(Switch01, HIGH); //
38         digitalWrite(Switch03, HIGH); //
39         delay(100); // Espera
40
41         while(1){
42             Memoria = analogRead(Velocidad);
43             Pwm = map(Memoria, 0, 1023, 0, 255);
44             analogWrite(Switch03, Pwm);
45             if( analogRead(Parada) ) {
46                 delay(100); // Espera
47                 digitalWrite(Switch01, LOW); //
48                 digitalWrite(Switch02, LOW); //
49                 digitalWrite(Switch03, LOW); //
50                 digitalWrite(Switch04, LOW); //
51                 delay(100); // Espera
52                 break;
53             }
54         }
55     }
56 }
```



```
57
58  if( analogRead(ArranqueAntiHorario) ) {  //
59      delay(100); // Espera
60      digitalWrite(Switch01, LOW);  //
61      digitalWrite(Switch03, LOW);  //
62      delay(1000); //spera
63      digitalWrite(Switch02, HIGH);  //
64      digitalWrite(Switch04, HIGH);  //
65      delay(100); // Espera
66      while(1){
67          Memoria = analogRead(Velocidad);
68          Pwm = map(Memoria, 0, 1023, 0, 255);
69          analogWrite(Switch04, Pwm);
70          if( analogRead(Parada) ) {
71              delay(100); // Espera
72              digitalWrite(Switch01, LOW);  //
73              digitalWrite(Switch02, LOW);  //
74              digitalWrite(Switch03, LOW);  //
75              digitalWrite(Switch04, LOW);  //
76              delay(100); // Espera
77              break;
78          }
79      }
80  }
```

```
81
82  if( analogRead(Parada) ) {  //
83      delay(100); // Espera
84      digitalWrite(Switch01, LOW);  //
85      digitalWrite(Switch02, LOW);  //
86      digitalWrite(Switch03, LOW);  //
87      digitalWrite(Switch04, LOW);  //
88      delay(100); // Espera
89  }
90
91  }
92
```

REFERENCIAS

Inventables (s.f.). *Cómo conectar un Mosfet de Potencia a un Microcontrolador*.
<https://www.inventable.eu/como-conectar-un-mosfet-a-un-microcontrolador/>

Inventables (2017, Set 4). *Los diodos 1N4000 y 1N5400*.
<https://www.inventable.eu/2017/09/04/los-diodos-1n4000/>

Orendain, S. (2020, Ago 18). *¿Cómo funciona un Diodo Zener?*
<https://circuitoslistos.com/como-functiona-un-diodo-zener/>

Llamas, L. (2016). *Controlar grandes cargas con Arduino y transistor MOSFET*.
<https://www.luisllamas.es/arduino-transistor-mosfet/>

Using FET's to Switch a load (2015. Dic 29). *Using FET's to Switch a load* (2015. Dic 29).
<https://arduino diy.wordpress.com/2015/12/29/using-fets-to-switch-a-load/>

Geek Factory (2013-2023). *Alimentar el Arduino: La guía definitiva*.
<https://www.geekfactory.mx/tutoriales-arduino/alimentar-el-arduino-la-guia-definitiva/>

Portal Mecatrónica (2020). *Puente H con Transistores E-Mosfet* [Video]. YouTube
<https://www.youtube.com/watch?v=85jYQuPLHDY>

Поделкин Самоделкин (2016, Jul 30). *ШИМ на ne 555* [Video]. YouTube.
<https://www.youtube.com/watch?v=SH4dEL0gAIY>
<https://radiokot.ru/lab/hardwork/67/04.jpg>

Opto Couplers (s.f.).
<https://www.talkingelectronics.com/ChipDataEbook-1d/html/OptoList.html#Works>

Arduino (s.f.). *Full Pinout*.
<https://content.arduino.cc/assets/A000066-full-pinout.pdf>