

En programación, una variable está formada por un espacio en el sistema de almacenaje (memoria principal de un ordenador) y un nombre (un identificador) que está asociado a ese espacio. Ese espacio contiene resto de información conocida o desconocida, es decir un valor.

En el sistema Arduino se pueden tener varios tipos de variables. Byte, int, long, float, char y array.

1- Con que podemos comparar una variable:

- a- Con un objeto
- b- Con una maleta
- c- Con un computador
- d- Con un amigo

Situación 1: *Se requiere desarrollar un sistema de control de peso para un ascensor, para esto se usa un módulo conversor de peso a voltios, el cual varia de 0 a 5 voltios proporcionalmente para 0 Toneladas a 5 toneladas.*

Las siguientes tres(3) preguntas, se harán teniendo en cuenta la Situación 1.

2- Se utiliza una variable int llamada elPeso, ¿cuál de las funciones sería la más apropiada para asignarle el valor del peso que recibe el módulo?

- a- digitalRead(pin)
- b- analogWrite(pin)
- c- digitalWrite(pin)
- d- analogRead(pin)

“...Lee el valor del pin analógico especificado. La placa Arduino contiene 6 canales (7 canales en placas MKR, 8 en Mini y Nano, 16 en Mega), convertidor de analógico a digital de 10 bits. Esto significa que mapeará voltajes de entrada entre 0 y 5 voltios en valores enteros entre 0 y 1023. Esto produce una resolución entre lecturas de: 5 voltios / 1024 unidades o, .0049 voltios (4.9 mV) por unidad...”

3- ¿Cuál de las siguientes condiciones sería la adecuada para detectar cuando el peso supera las 3 toneladas?

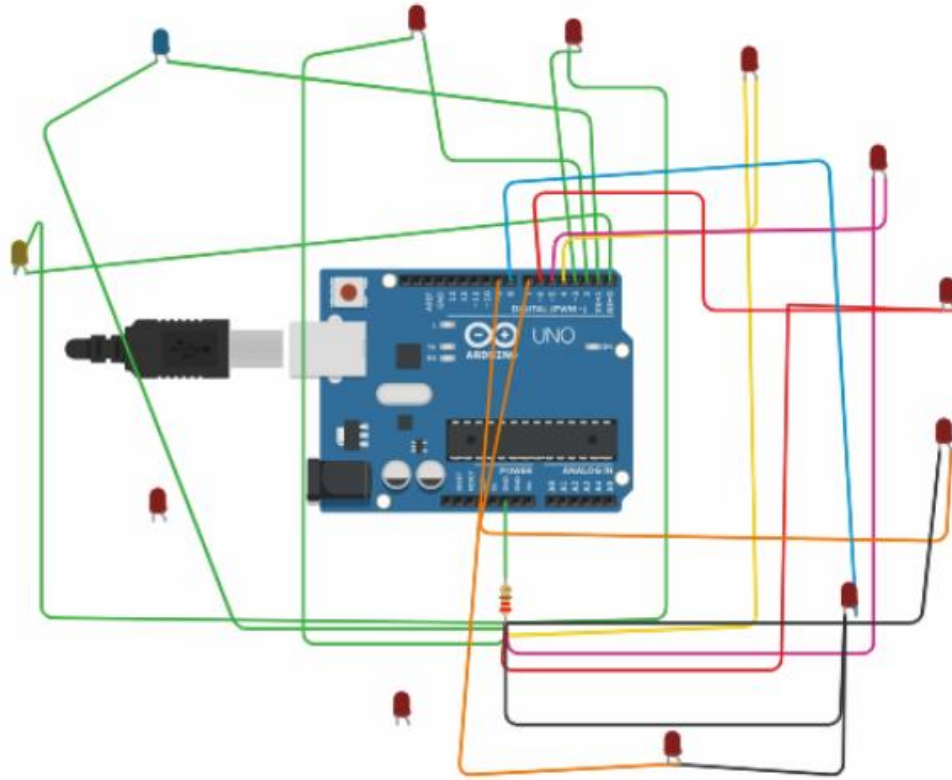
- a- (elPeso > 3*1024 /5)
- b- (elPeso == 3)
- c- (elPeso > 3)
- d- (elPeso == 3*1024 /5)

4-Cuál de las siguientes instrucciones se deberá introducir en la función loop() para activar una alarma cuando se cumpla la condición

- a- while (condición){}
- b- for (condición){}
- c- if(condición){}

d- switch(condición){}

Teniendo el siguiente circuito, en el cual están conectados leds desde el pin 0 al pin 9 del módulo DIGITAL



```
loop(){
    for (i=3; i<6; i++){
        digitalWrite(i,HIGH);
    }
}
```

5- Según el montaje y código anterior, después de un segundo de iniciado el programa:

- a- Los pines digitales 3, 4, 5 y 6 están prendidos
- b- Los pines digitales 2, 3, 4, 5 y 6 están prendidos
- c- Los pines digitales 3, 4 y 5 están prendidos
- d- Los pines digitales 3, 4, 5, 6 y 7 están prendidos

El operador NO permite invertir el valor de los bits de una variable de esta forma convierte 0 en 1 y 1 a 0 así:

```
int a = 103; // binary: 0000000001100111
```

```
int b = ~a; // binary: 1111111110011000
```

```

bool Prendido = 0;

int Tiempo = 5000;

...

loop(){

    while (Tiempo >= 1000){

        digitalWrite(LED_BUILDING, Prendido);

        delay(Tiempo);

        Tiempo = Tiempo - 1000;

        Prendido = ~ Prendido;

    }

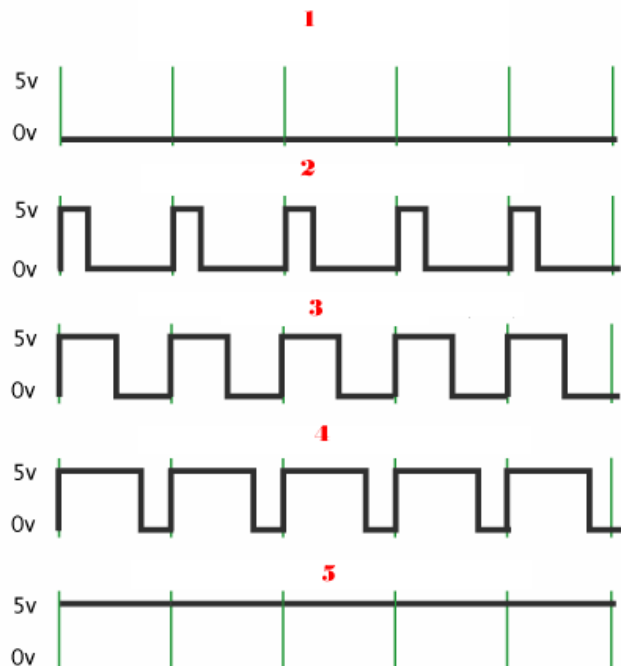
}

```

- 6- ¿De acuerdo al código anterior, el LED interno queda prendido después de 30 segundos?
- a- Verdadero
 - b- Falso
 - c- No se puede determinar
 - d- Depende del Tiempo

La modulación por ancho de pulso, o PWM, es una técnica para obtener resultados analógicos con medios digitales. El control digital se utiliza para crear una onda cuadrada, una señal activada y desactivada. Este patrón de encendido y apagado puede simular voltajes entre encendido total (5 voltios) y apagado (0 voltios) cambiando la parte del tiempo que la señal pasa en comparación con el tiempo que la señal pasa. La duración de "on time" se denomina ancho de pulso. Para obtener valores analógicos variables, cambia o modula ese ancho de pulso. Si repite este patrón de encendido / apagado lo suficientemente rápido con un LED, por ejemplo, el resultado es como si la señal fuera un voltaje constante entre 0 y 5v que controla el brillo del LED.

En el siguiente gráfico, las líneas verdes representan un período de tiempo regular. Esta duración o período es el inverso de la frecuencia PWM. En otras palabras, con la frecuencia PWM de Arduino a aproximadamente 500Hz, las líneas verdes medirían 2 milisegundos cada una. Una llamada a `analogWrite()` está en una escala de 0 a 255, de modo que `analogWrite(255)` solicita un ciclo de trabajo del 100% (siempre activado), y `analogWrite(127)` es un ciclo de trabajo del 50% (en la mitad del tiempo) para ejemplo.



7- De acuerdo a lo expuesto anteriormente, cual grafica representa la onda generada por la instrucción `analogWrite(125)`.

- a- 1
- b- 2
- c- 3
- d- 4

8- De acuerdo a lo expuesto anteriormente, cual grafica representa la onda generada por la instrucción `analogWrite(62)`.

- a- 5
- b- 4
- c- 3
- d- 2

Teniendo las siguientes líneas de código

```
char Pepito[] = "desjuiciado";
char Juanito[] = "inquieto";
loop(){
    if (Juanito[]="inquieto" || Pepito[]="juicioso"){
        digitalWrite(LED_BUILTIN,0);
    }
    else{
        digitalWrite(LED_BUILTIN,1);
    }
}
```

9- El led se enciende si:

- a- `Pepito[] = "desjuiciado";char Juanito[] = "inquieto";`
- b- `Pepito[] = "juicioso";char Juanito[] = "quieto";`
- c- `Pepito[] = "juicioso";char Juanito[] = "inquieto";`
- d- `Pepito[] = "desjuiciado";char Juanito[] = "quieto";`

La definición de la función `random` es

`random(min, max)`

10- Esto significa que:

- a- `min` y `max` son los parámetros que definen el rango en el cual se desea el valor de retorno.
- b- Es una función que debe tener dos variables
- c- Es una variable aleatoria
- d- Es una función para sacar un numero de 1 a 256

