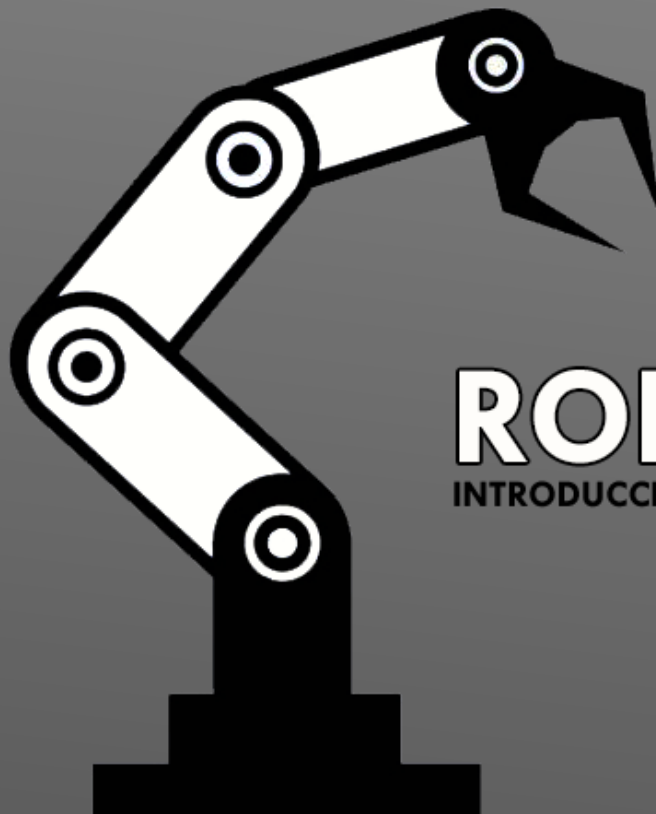




UNED

UNIVERSIDAD ESTATAL A DISTANCIA

Institución Benemérita de la Educación y la Cultura



ROBÓTICA

INTRODUCCIÓN Y PRINCIPIOS BÁSICOS

MOTORES

MODALIDAD: INDIVIDUAL

VALOR PORCENTUAL: 5%

PUNTOS: 20 PTOS

OBJETIVO: FAMILIARIZARSE CON LA PROGRAMACIÓN EN EL AMBIENTE ARDUINO MEDIANTE EL USO DE FUNCIONES, LUCES Y SENSORES

RECURSOS NECESARIOS PARA REALIZAR LA ACTIVIDAD:

- TENER INSTALADO EL PROGRAMA ARDUINO EN SU PC
- 4 CABLES FEMALE - FEMALE
- 1 CABLE USB MINI B
- EL ROBOT COMPLETO

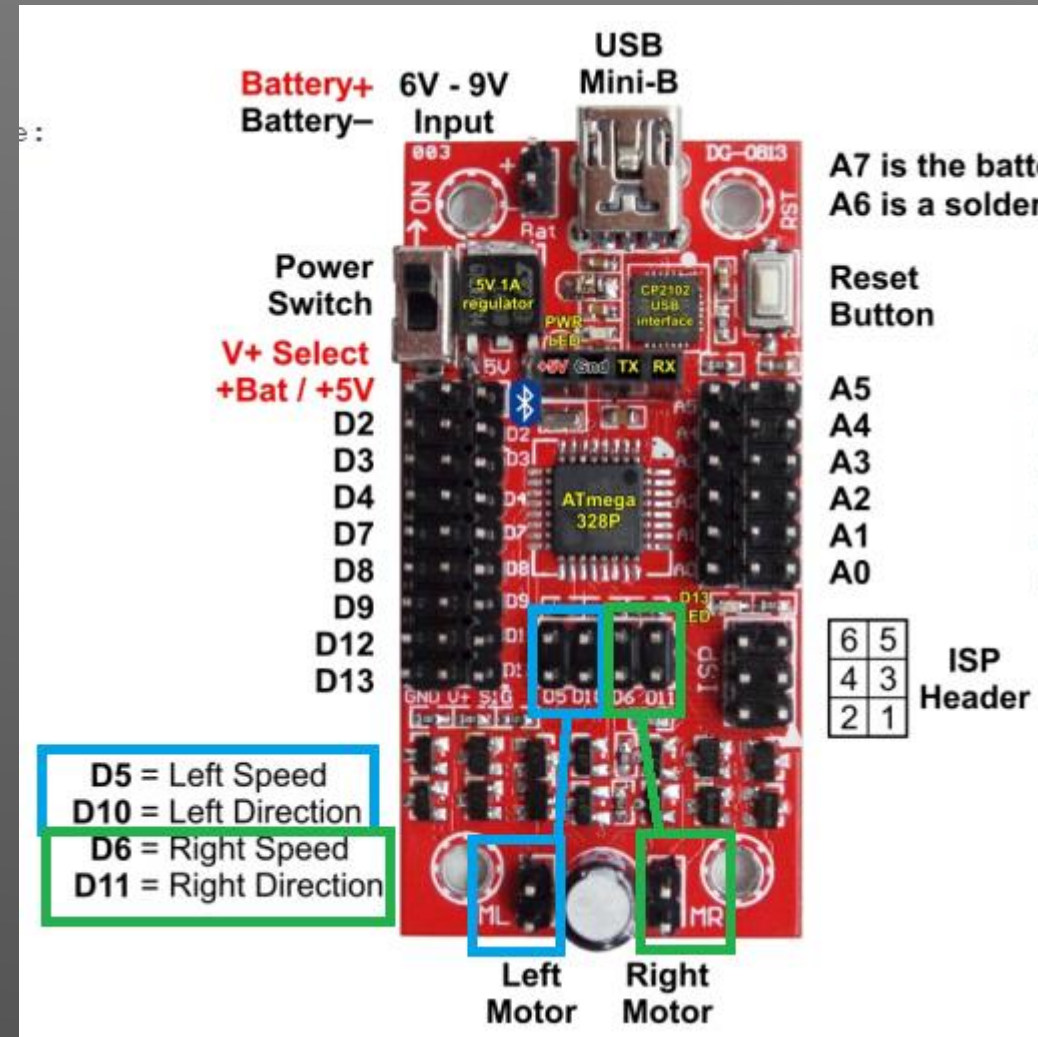
INSTRUCCIONES PARA REALIZAR LA ACTIVIDAD :

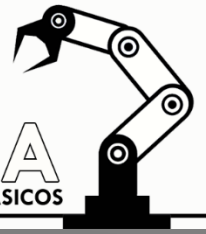
- LEA PRIMERO TODA LA PRÁCTICA.
- EJECUTE LOS PASOS DE ACUERDO A LO SOLICITADO.
- ENTREGUE LAS EVIDENCIAS DE APRENDIZAJE (FOTOGRAFÍAS DEL TRABAJO HECHO) POR MEDIO DEL ENLACE DE ENTREGA.

Motores

La placa S4A EDU Robotic controller. Tiene reservados, gracias a cuatro jumpers los pines D5 y D10 para controlar el motor izquierdo y los pines D6 y D11 para controlar el motor derecho.

Crearemos un nuevo programa de Arduino usando el IDE, lo guardaremos con el nombre laboratorio7, dentro de una carpeta con el mismo nombre.





Los comentarios de los códigos no es necesario que los copien, su función es que ustedes sepan que hace cada línea, los comentarios se pueden escribir después de dos barras inclinadas.

Laboratorio7

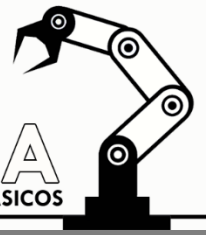
```
1 int velocidadDerecha=100; //Esta variable contiene el valor utilizado en el control de velocidad del motor derecho .  
2 int velocidadIzquierda=100; //Esta variable contiene el valor utilizado en el control de velocidad del motor izquierdo.
```

Declararemos dos variables para manejar el valor de la potencia que se le da al motor. La primera variable se llamará velocidadDerecha con un valor de 100, y una segunda variable global llamada velocidadIzquierda, también con una potencia de 100

La potencia de un motor DC, puede establecerse desde el valor 0, que sería detenido a 255, que es la velocidad máxima.

Así que en la línea 1 se declara que la variable velocidadDerecha valdrá 100

Así que en la línea 2 se declara que la variable velocidadIzquierda valdrá 100 también.



En la función `void setup` declararemos la función que cumplirán los pines de cada puerto, declararemos que todos son puertos de salida (`OUTPUT`)

```
3
4 void setup() {
5   pinMode(10,OUTPUT); //El pin 10 controla la dirección en que rota el motor derecho
6   pinMode(5,OUTPUT);  //El pin 5 controla la velocidad del motor derecho
7   pinMode(11,OUTPUT); //El pin 11 controla la dirección en que rota el motor izquierdo
8   pinMode(6,OUTPUT);  //El pin 6 controla la velocidad del motor izquierdo
9   pinMode(12, OUTPUT); //En el pin 12 conectaremos una luz led
10  pinMode(13, OUTPUT); //En el pin 13 conectaremos una luz led
11  pinMode(9, OUTPUT);  //En el pin 9 conectaremos una luz led
12 }
13 void loop() {
```

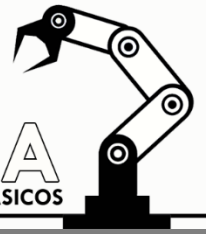


En la función `void loop ()`; llamaremos a las funciones como adelante, reversa, giro (Por nosotros mismos) que son:

Estas funciones las declararemos después de la función `void loop()`;

Por el momento solo invocaremos a la función `adelante()`;

```
13 void loop() {  
14   adelante();  
15 }
```



Ahora declararemos la función que permitirá que el robot camine hacia adelante,

```
16
17 //Configura los motores para ir hacia adelante
18 void adelante()
19 {
20     digitalWrite(10, HIGH); // HIGH hará que la tracción del motor derecho sea hacia adelante
21     digitalWrite(11, HIGH); // HIGH hará que la tracción del motor izquierdo sea hacia adelante
22     analogWrite(5, velocidadDerecha);
23     analogWrite(6, velocidadIzquierda);
24     digitalWrite(12, HIGH); // enciende el led del pin 12, el verde
25     digitalWrite(13, LOW); // apaga el led del pin 13, el rojo
26     digitalWrite(9, LOW); // apaga el led del pin 13, el rojo
27 }
28
```

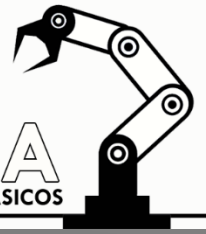
Lo que hace la función “adelante” es lo siguiente:

En la línea 20 se le asigna al pin 10 que controla la dirección de giro del motor para que gire hacia adelante (HIGH hace que gire para adelante y LOW hará que gire para atrás)

Lo mismo se hace en la línea 21, pero con el motor izquierdo.

En las líneas 22 y 23 se asigna al pin que manipula la velocidad, el valor de las variables que se definió al principio del programa (velocidadDerecha y velocidadIzquierda).

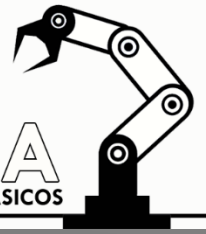
En la línea 24 se enciende la luz conectada al pin 12 y se apagan las luces de los pines 13 y 9.



Podemos guardar los cambios hechos al programa motores y subirlos a la placa S4A.

Sí nuestro robot cuenta con sus 4 baterías AA, lo podremos desconectar de la PC, y encenderlo por medio del interruptor de encendido apagado. Coloque el robot en el suelo y espere unos segundos.

Si ha escrito bien el programa, su robot deberá, después de unos segundos en el suelo, empezar a cumplir su programa, caminando hacia adelante, hasta que encuentre un obstáculo que le impida continuar.

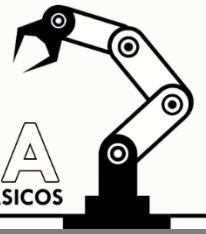


Escribiremos una función más después de la función adelante, llamada reversa, de tal manera que el robot pueda avanzar y retroceder.

```
29 void reversa()  
30 {  
31     digitalWrite(10, LOW); // LOW hará que la tracción del motor derecho sea hacia atrás  
32     digitalWrite(11, LOW); // LOW hará que la tracción del motor izquierdo sea hacia atrás  
33     analogWrite(5, velocidadDerecha);  
34     analogWrite(6, velocidadIzquierda);  
35     digitalWrite(12, LOW); // apaga el led del pin 12, el verde  
36     digitalWrite(13, HIGH); // enciende el led del pin 13, el rojo  
37     digitalWrite(9, LOW); // apaga el led del pin 13, el rojo  
38 }
```

Como podemos observar el prácticamente el mismo programa, solo cambiamos los valores de los pines 10 y 11 a LOW, invirtiendo su giro.

También, y solo como para comprobar que la función que se está ejecutando es reversa(), apagamos la luz led 12 y encendemos la 13



Para que el robot ejecute las dos acciones, debemos de invocar la función `reversa()` en la función `loop()`, añadiremos un tiempo de ejecución por medio de la sentencia `delay`, haremos que el robot camine hacia adelante 3 segundos y retroceda 2 segundos, de la siguiente manera:

```
13 void loop() {  
14   adelante();  
15   delay(3000);  
16   reversa();  
17   delay(2000);  
18 }
```

Podemos subir el programa a la placa, desconectar el robot y probar el funcionamiento del programa.

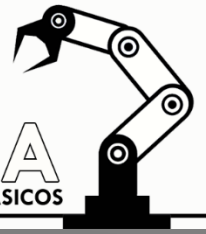


Agregaremos una función más: Que el robot gire a la derecha, esto será muy sencillo, ya que escribiremos una función que ponga un motor que gire hacia adelante y el otro hacía atrás, esta función la correremos por un segundo, ya que si lo hacemos por más tiempo el robot dará vueltas sobre si mismo

Para que gire a la derecha pondremos el pin 10 en HIGH y el pin derecho en LOW

A la función la llamaremos giro()

```
46 void giro()
47 {
48     digitalWrite(10, HIGH); // HIGH hará que la tracción del motor derecho sea hacia adelante
49     digitalWrite(11, LOW);  // LOW hará que la tracción del motor izquierdo sea hacia atrás
50     analogWrite(5, velocidadDerecha);
51     analogWrite(6, velocidadIzquierda);
52     digitalWrite(12, LOW);   // apaga el led del pin 12, el verde
53     digitalWrite(13, LOW);   // apaga el led del pin 13, el rojo
54     digitalWrite(9, HIGH);   // enciende el led del pin 13, el rojo
55 }
```



Y en el loop() invocaremos las funciones en el siguiente orden: Que camine para adelante 3 segundos, gire a la derecha 1 segundo y retroceda por 2 segundos.

```
13 void loop() {  
14   adelante();  
15   delay(3000);  
16   giro();  
17   delay(1000);  
18   reversa();  
19   delay(2000);  
20 }
```

Podemos subir el programa a la placa, desconectar el robot y probar el funcionamiento del programa.

Crearemos una última función, que hará que el robot se detenga. Para lograrlo cambiaremos los valores de la variables globales a 0.

Aprovecharemos también para encender todas las luces led

```
60 void detener() {  
61   velocidadDerecha=0;  
62   velocidadIzquierda=0;  
63   analogWrite(5, velocidadDerecha);  
64   analogWrite(6, velocidadIzquierda);  
65   digitalWrite(9, HIGH);  
66   digitalWrite(12, LOW);  
67   digitalWrite(13, LOW);  
68 }
```

En el loop, haremos los siguientes cambios:

- Regresamos el valor de las variables globales a 100, por que si no lo hacemos, ahora el valor de las variables será de cero y ya no podría caminar.
- Luego haremos que camine para adelante 3 segundos
- Gire a la derecha por 1 segundo
- Retroceda por dos segundos
- Se detenga por dos segundos.


Como todas las funciones se invocan desde el loop, una vez terminado el ciclo, el mismo volverá a repetirse.

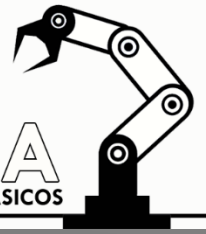
```
13 void loop() {  
14   velocidadDerecha=100;  
15   velocidadIzquierda=100;  
16   adelante();  
17   delay(3000);  
18   giro();  
19   delay(1000);  
20   reversa();  
21   delay(2000);  
22   detener();  
23   delay(2000);  
24 }
```

Podemos subir el programa a la placa, desconectar el robot, encender el interruptor y probar el funcionamiento del programa.



Nuestro robot no es inteligente aún, pero ya sabemos como hacer para que camine hacía adelante, atrás, gire y se detenga.





Con cualquier cámara, incluso un teléfono celular, tome un vídeo de su robot ejecutando el programa que escribimos.

Suba el vídeo a la plataforma por medio del enlace llamado práctica 7

Suba también el la carpeta laboratorio7 con el programa de Arduino, comprimida en zip para su revisión.



Laboratorio práctico 7 Valor 5%		
Nombre del estudiante:		
Rubro	Valor	Ptos. Obtenidos
Entrega		
Entrega el vídeo y la carpeta zip solicitados en la fecha indicada.	5pts	
Utiliza el buzón correspondiente para subir su tarea.	2 pts	
Cumple los Objetivos	5	
El robot cumple todas las funciones: - adelante(); - giro(); - reversa(); - detener(); En los tiempos solicitados.	8	
Total	20	
Comentarios:		