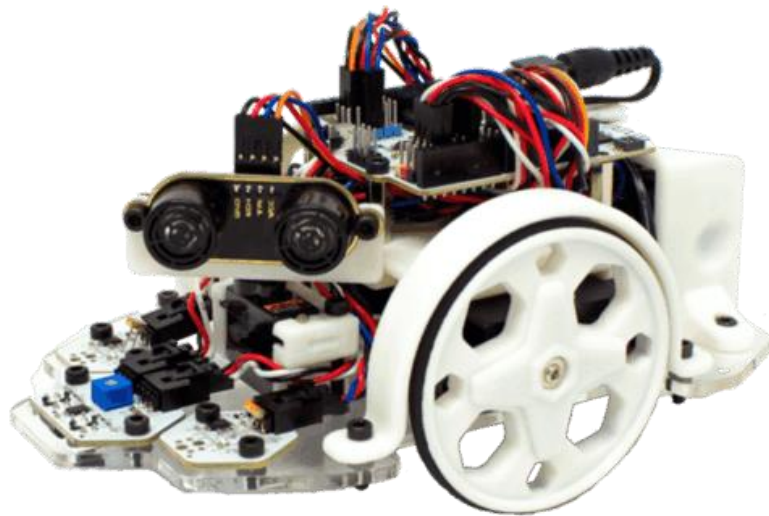




ROBOT EXPLORADOR

Grupo de Laboratorio D1



8 DE MAYO DE 2019

Introducción

La práctica consiste en programar un robot bq con Arduino Zero, para que haga diversas funcionalidades. Se trata de simular un robot explorador que vaya moviéndose a lo largo de una superficie principalmente blanca, con el fin de encontrar áreas más oscuras que indican la presencia de minerales.

Un archivo nos muestra las funcionalidades que este debe desempeñar. Las más importantes como hemos mostrado anteriormente son las de encontrar zonas más oscuras sobre el terreno, pararse, emitir un sonido durante cinco segundos y reanudar la marcha. Además, de comprobar el almacenamiento de las variables y el código en la memoria. También, nos presentan unas funciones extras que se pueden llevar a cabo, a través de diversos sensores y actuadores, como la sensibilidad a la luz, detener el robot una vez haya alcanzado un tiempo de 3 minutos de exploración, emplear interrupciones para los infrarrojos, detectar objetos que estén situados a una distancia menor a 5 cm o desarrollar diferentes tipos de trayectorias durante la expedición.

➤ Sensores de entrada

- **Ultrasonidos**, consiste en un sensor que emite una señal de alta frecuencia y un receptor la recibe cuando esta rebota en algún objeto, para saber la distancia a la que se encuentra.
- **Infrarrojo**, este sensor consta de un LED emisor que emite infrarrojos y un fotodiodo que capta la señal reflejada. Este sensor es capaz de distinguir entre negro y blanco según la cantidad de luz reflejada.
- **Luz**, este sensor se basa en el uso de una resistencia LDR, cuyo valor Óhmico depende de la luz que incide en ella. Por tanto, es capaz de proporcionar una señal analógica proporcional a la luz que incide en ella.

➤ Sensores de salida o actuadores

- **Zumbador**, es un tipo de actuador que permite generar sonidos mediante el uso de una salida PWM con la que se puede fijar la frecuencia del sonido a generar
- **Servomotor**, para mover el robot se emplean dos servomotores de rotación continua que permiten regular la velocidad y sentido de giro de cada rueda del mismo de forma independiente.
- **Miniservo**, el robot emplea un miniservo para orientar el sensor de ultrasonidos. Este tipo de servomotor, a diferencia de un servomotor de rotación continua, permite posicionar el eje en una posición entre 0 y 180 grados.

Funciones

A continuación, se van a citar las funciones que le hemos introducido al robot y se hará una explicación de cada una de ellas:

- **Sensibilidad a la luz:** con esta funcionalidad el robot es capaz de detectar si hay más luz por un lado u otro y dependiendo de la luz que entre, este realizará las funciones o no. Esto será gracias a los sensores de luz, los cuales los hemos ajustado en un umbral de 50 (el máximo es 1024). Si ambos sensores detectan más de 50, no se verá afectado, es decir, realizará las funciones. Sin embargo, si el umbral de luz es menor a 50 el robot se mantendrá parado. Para la realización de esta función se ha hecho uso de condicionales comprobando cuánto vale el valor recogido por los sensores.
- **Detección de objetos:** se realiza con sensores de ultrasonidos. Gracias a esta funcionalidad el robot es capaz de detectar la distancia a la que están los objetos de él. En este caso cuando un objeto se encuentra a menos de 5 cm de él, pita, se para y transcurridos 3 segundos, deja de sonar y continúa con su camino. Gracias a esto, es capaz de no chocarse con los objetos que se encuentren en su camino.
- **Detección de color negro:** es capaz gracias al sensor infrarrojo. Con esta funcionalidad el robot es capaz de detenerse ante una mancha negra en el suelo y pitar durante 5 segundos. Transcurrido este tiempo continúa la marcha y deja de sonar en busca de más zonas negras.
- **Duración máxima de 3 minutos:** gracias al uso de un bucle condicional "if" y a la función mili, el robot es capaz de hacer una exploración de 3 minutos hasta quedarse parado por completo.
- **Reproducción de sonidos:** El robot reproduce los sonidos que le hemos introducido en el momento que queremos que se reproduzcan. Se realiza con un zumbador. Hemos introducido las notas y las duraciones de estas en matrices distintas para reproducirlas con un bucle usando las funciones 'tone' y 'noTone'. Hemos introducido la primera melodía en el 'setup' para que se reproduzca al encender el robot y otra melodía que se reproducirá antes de dar media vuelta al tapar los sensores de luz.

Informe del Proyecto

1. Funcionalidades del Proyecto:

Funcionalidad	Completada (Sí/No)
Funcionalidad básica 1: movimiento exploratorio, detección de zonas oscuras y sonido de 3 kHz.	Sí
Funcionalidad básica 2: direcciones de memoria y tamaño de las variables, tamaño del programa y tamaño de las secciones de memoria (uso de AVR-size).	Sí
Funcionalidad avanzada 3: paro cuando el nivel de luz es bajo.	Sí
Funcionalidad avanzada 4: detener exploración después de 3 minutos y hacer sonar una melodía.	Sí
Funcionalidad avanzada 5: uso de interrupciones para los sensores de infrarrojo.	
Funcionalidad avanzada 6: detectar objetos y emitir un sonido de 500 Hz cuando la distancia sea menor que 5 cm.	Sí
Funcionalidad avanzada 7: uso de diferentes trayectorias de exploración	Sí
Funcionalidades adicionales	Melodía de espera cuando no hay luz

2. Variables y programa en memoria (Funcionalidad básica 2)

Nombre Variable	Tipo	Tamaño (bytes)	Global (Sí/No)	Dirección de memoria
Luz_left	int	4 bytes	Sí	200074D8
Luz_right	int	4 bytes	Sí	200074DC
Servo_izquierdo	int	4 bytes	Sí	2000755C
Servo_derecho	int	4 bytes	Sí	200074E4
IR_left	int	4 bytes	Sí	200074E0
IR_right	int	4 bytes	Sí	200074D4
NEGRO	int	4 bytes	Sí	20007558
Buzzpin	int	4 bytes	Sí	200074E8

Tamaño total del programa y % de memoria ocupada	23.528 y 8%
Tamaño de la sección .text	23528 bytes
Tamaño de la sección .data	448 bytes
Tamaño de la sección .bss	3500 bytes

3. Descripción general del proyecto

- En la función **setup ()** lo que hacemos es iniciar los sensores, es decir, configuramos los pines donde se conectan dichos sensores. Esta configuración la realizamos mediante la función `pinMode` (variable del sensor, modo(entrada/salida)). Y, además, inicializamos la comunicación con el puerto serie a 9600 baudios.
- La función **loop ()**, básicamente actúa como un bucle que permite al programa ejecutarse. Es decir, contiene el código que se ejecutará continuamente (lectura de

entradas, activación de salidas, etc). En nuestro caso, hemos metido dos “if” anidados que contienen todo lo necesario para que el robot desempeñe las funciones que se piden en la práctica. Contenidos de dicha función:

- El primer “if”, solamente se ejecuta si la variable tiempoTotal es mayor a 180000(3 minutos en milisegundos, ya que la función millis() se expresa en milisegundos). En este caso el robot se pararía y emitiría una melodía. Dicha función permite que el robot solo efectúe un recorrido de 3 minutos, parándose cuando dicho tiempo se sobrepase.
- El segundo “if” se ejecuta mientras el umbral de luz sea mayor a 50, ya que como pone en la práctica el robot explorador solo recarga su batería cuando hay luz solar. Si se cumple, el robot inicia la lectura de datos con infrarrojos y su trayectoria. Si el valor del infrarrojo es igual a la variable negro, el robot se para, emite un sonido durante 5 segundos y luego reanuda la marcha. A parte de esto, el robot incorpora un sensor ultrasonidos, con él va calculando las distancias de los objetos que lo rodean. Si esta distancia es menor a 5 cm, para, emite un sonido durante 3 segundos y reanuda la marcha.

En el caso de que no hubiera luz el robot se mantendría quieto, sin leer ningún dato y emitiría un sonido que nos indicaría que la luz es escasa.

4. Descripción de funcionalidades adicionales (no se refiere a las funcionalidades 1 a 7).

Hemos añadido una melodía al robot cuando este se encuentra en espera, porque el sensor de luz no detecta el umbral de luz suficiente para funcionar.

5. Problemas más relevantes encontrados durante la realización del proyecto.

Algunos robots tenían los sensores defectuosos y a la hora de probar el código se nos hacía difícil saber donde estaba el fallo si en el código o en el robot. Por lo demás, con los ejemplos de campus virtual, hemos podido guiarnos para la implementación del código.