



INSTITUTO TECNOLÓGICO DE COSTA RICA

ÁREA DE INGENIERÍA EN COMPUTADORES

ARQUITECTURA DE COMPUTADORES I

Proyecto II: Informe

Estudiantes:

Luis BARBOZA A.
Malcolm DAVIS S.
Sebastián VÍQUEZ R.

Profesor:

Ing. Fabián ZAMORA

30 de noviembre de 2017

Índice

1. Comparación de propuestas	2
2. Requerimientos del sistema	4
3. Tecnologías utilizadas	6
3.1. Microcontrolador	6
3.2. Sensores	6
3.2.1. Infrarrojo	6
3.2.2. Módulo RFID	6
3.2.3. Distancia	7
3.3. Sistema móvil	7
3.4. Comunicación	7
4. Metodología de diseño utilizada	8
4.1. Hardware	8
4.2. Software	8
5. Conclusiones y lecciones aprendidas	9

1. Comparación de propuestas

Para este proyecto se analizaron dos propuestas. La primera consiste en un robot capaz de clasificar la basura según el tipo de material y la otra idea es un robot de un warehouse, donde tiene que llevar paquetes a un lugar en específico según la identificación que posea.

Ambas propuestas representan problemas de la vida actual, donde la primera es evitar que personas depositen la basura en un lugar equivocado y así reciclar la mayor cantidad de material. La segunda alternativa corresponde a un mayor orden y control de tantos paquetes que pueden haber un warehouse. Saber dónde deben dirigirse es una gran tarea y es por eso que un robot de este tipo sería de gran ayuda.

Para este proyecto se decidió escoger el robot warehouse por las siguientes razones:

Precio

Aunque el precio de esta opción era el más elevado, hay muchos materiales que no se tenían que comprar porque alguno de los miembros del grupo ya los poseía. De toda la lista de componentes o partes que se necesitaban para la construcción únicamente se tuvieron que comprar los sensores infrarrojos y los componentes para la maqueta donde el carrito se va a desplazar a realizar las entregas de los paquetes.

Para la otra alternativa se debían comprar todos los sensores (infrarrojo, inductivo y fuerza). Al sumar estos tres componentes el precio iba a ser mayor que la opción del robot del warehouse.

Implementación

En este aspecto se analizó la dificultad para implementar el proyecto. En el caso del basurero se debe tomar en cuenta cada uno de los sensores para analizar el tipo de material que se está ingresando. Las muestras que se van a comparar son plástico, papel y metal. Para el caso del plástico se asumió que su material fuera traslúcido, pero en el mercado se encuentran envases plásticos que no son traslúcidos, por lo que no cumpliría con esta condición y podrían generar errores si se depositan en un envase incorrecto. Al igual que con el plástico, el metal podía tener cierto problema para ser analizado y en caso de que no haya pasado por ninguno de los sensores, la basura se iba a depositar en el de papel, por lo que presenta un riesgo para el manejo de desechos.

En el caso del robot para el warehouse, los sensores que tiene aceptan una medida no tan exacta por lo que se puede manejar un error. Para la lectura del código en cada paquete, se debe tomar en cuenta la distancia en la que se encuentra y se va a lograr una interpretación correcta del identificador del dispositivo. El otro sensor corresponde a el reconocimiento de la línea para circular por ella, por lo que consiste en estar haciendo lecturas de sensor para analizar sobre qué color está.

En términos de implementación el basurero clasificador implica un enfoque distinto para determinar el tipo de material y no sucedan errores cuando se vaya separar un desecho. En cambio, el de warehouse tiene un margen un poco más amplio para errores en lectura y no va a perjudicar el funcionamiento.

Construcción

El aspecto de la construcción es un factor muy importante porque se debe tomar en cuenta para lograr realizar una estructura capaz de crear un correcto funcionamiento del robot.

Con el basurero clasificador se debe tener una configuración de manera que todos los sensores se deben colocar en el mismo lugar. La razón de esto es que todas las mediciones las realiza sobre la entrada del sistema. El nivel de complejidad aumenta para que los diferentes componentes no choquen y generen conflictos. Además, se tiene que tener en cuenta que puede existir la posibilidad que, depende de la colocación del sensor, la medición de algún sensor pueda interferir en la otra.

Al contrario, el robot de warehouse permite la colocación de los sensores en distintas ubicaciones del robot. Para el robot se ideó un carrito para transportar las cargas, por lo que los sensores infrarrojos se pueden colocar al frente de carro y el de lectura encima donde se va a encontrar la carga.

En este caso la dificultad de la estructura del robot genera un aspecto más para analizar entre las dos propuestas. Es importante decir que se escogió el robot warehouse para lograr un correcto funcionamiento y no que ocurran errores entre los diferentes sensores.

2. Requerimientos del sistema

Para el proyecto se generó una lista de requerimientos que se debían cumplir en el desarrollo, lo cuales se explican a continuación:

Programación basada en microcontrolador

El microcontrolador escogido para el proyecto es el ATmega328P. La razón de esta elección es porque ya se poseía un Arduino y era más sencillo extraerle el microcontrolador. Para la parte programada se realizó el código funcional para controlar el robot y luego se quemó toda la lógica al microcontrolador usando el Arduino como medio para pasar toda la información. Una vez realizado esto, se procede a desmontar el microcontrolador y está listo para su uso.

Sensado

Los sensores utilizados son: infrarrojo, RFID y ultrasónico.

Para analizar el camino que debe seguir, se hizo uso de tres sensores colocados al frente del carrito. La cantidad es así porque hay tres posibilidades que puede tomar: doblar a la izquierda, seguir directo o doblar a la derecha. Un sensor es utilizado para cada una de estas alternativas.

El RFID corresponde a un módulo que va a leer las etiquetas que van a tener los diferentes paquetes para asociar la estación donde se va a colocar dicho elemento. El identificador se va a encontrar en forma de llavero pegado al paquete.

Por último, el ultrasónico se utiliza para parar el movimiento del carro en caso de que exista un obstáculo. La idea es que cuando esto ocurra, el personal encargado debe mover la pieza que está impidiendo el paso. Hasta que no se quite la pieza, el carro no se va a mover.

Movimiento

El movimiento es simple ya que es un carro. Posee la funcionalidad de moverse hacia adelante, doblar a la izquierda y a la derecha. Además, cuando el paquete haya sido entregado en la estación correspondiente, el carro debe dar un giro de 180 grados para devolverse a la base y recibir el siguiente paquete.

Esto se logra mediante motores DC que permiten el movimiento de las ruedas y así lograr el objetivo de desplazarse por la superficie.

Comunicación Bluetooth o WiFi

En este caso se eligió red WiFi por lo que se hizo necesario el uso de un módulo para este objetivo.

Recibir mensajes

El robot debe ser capaz de recibir mensajes por medio de la red. La idea es que pueda aceptar señales en caso de emergencia. La instrucción que puede obtener es 'regresar a la base'. Esta función se usa en caso de que el carro, sin llegar a la estación, quiera volver a la base donde se reparten los diferentes paquetes. Esto sucede para el caso de que se haya detectado un error y se requiere que no se entregue la información en el destino.

Enviar mensajes

El robot es capaz de enviar mensajes hacia la aplicación de escritorio con el fin de ampliar las estadísticas de los paquetes enviados hacia las diferentes estaciones. Por lo que cada vez que llega a un destino debe remitir a la aplicación, la estación que recibió el paquete correspondiente.

Aplicación de escritorio

La aplicación de escritorio, realizada en Processing, corresponde a un módulo capaz de enviar y recibir mensajes para controlar el funcionamiento del robot. Envía la señal de que el robot se debe regresar a la base en caso de que haya ocurrido un error con el paquete solicitado. Y permite recibir cada vez que un paquete es depositado en una estación, con el fin de contabilizar la cantidad de paquetes que se han repartido.

3. Tecnologías utilizadas

3.1. Microcontrolador

Opción	Características	Conclusión
ATmega328P	Microcontrolador de 8 bits. Basado en RISC. Memoria flash de 16 kB. 26 como número máximo de pines.	Se eligió esta opción por ser más conocida para el grupo por el hecho de encontrarse en el Arduino. Conocer una tecnología conlleva más tiempo, por lo que se decidió por este microcontrolador.
PIC16x84	Microcontrolador de 8 bits. Contiene 18 pines. Memoria flash de 1 kB.	Esta opción no era tan convincente por el hecho de que se tenía que buscar uno de estos por ser una tecnología no conocida por el grupo.

3.2. Sensores

3.2.1. Infrarrojo

Opción	Características	Conclusión
CNY70	Muy utilizado para seguidores de línea. Posee filtro de bloqueo a la luz del día. Rango de operación de 0 a 5mm. Puede utilizarse como entrada analógica o digital.	Componente económico que se ajusta a las necesidades para implementar el seguidor de línea y con la ventaja de tener un filtro de bloqueo de luz.
TCRT5000	Rango de detección de 0 a 8mm. Posee filtro de bloqueo a la luz del día. Tiene LED de indicador de estado.	Aunque se consideró usarlo por las características similares al CNY70, su precio era ligeramente más caro.
RGB	Detecta qué color tiene el carril. Con esta idea se pueden realizar carriles de diferentes colores para que el robot pueda seguirla.	Aunque es una buena opción para detectar más fácil el destino del paquete, presenta varios problemas por las condiciones donde se pruebe el sistema. Ya que si está muy iluminado puede afectar el valor del color y tomar otra ruta.

3.2.2. Módulo RFID

Opción	Características	Conclusión
RDM3600	Protocolo UART. Compatible con gran cantidad de microcontroladores. Rango de operación cercano a los 50mm	Ya que se contaba con este módulo, y al conocimiento sobre él, se procedió a utilizarlo en la implementación del proyecto.
MFRC-522	Se comunica por medio de SPI. Funciona en modo lectura y escritura de tags de RFID en una distancia de hasta 4cm. Buffer FIFO de 64 bytes. Amplio rango de temperatura de trabajo.	Debido a que se contaba con el RDM3600 y a su precio ligeramente elevado esta opción se descartó.

3.2.3. Distancia

Opción	Características	Conclusión
HC-SR04	Sensor ultrasónico. Rango de medición de 2cm a 4m. Tensión requerida 5V. Señal de salida compatible con lógica TTL. Económico.	Debido a su precio relativamente cómodo y a los rangos de detección de objetos se procedió a utilizarlo.
VCNL4010	Usado para distancias cortas, no mayores a 20cm. Rango de tensión de 3.3V a 5V. Fácil de usar con microcontroladores que poseen I2C.	Tiene un costo más elevado que el ultrasónico, por lo que se descartó.

3.3. Sistema móvil

Opción	Características	Conclusión
Chasis con motores y ruedas	Una sola pieza que incluye dos motores DC y dos ruedas. Con mecanismo para girar a la izquierda y derecha.	Aunque el precio es caro, debido a que ya se contaba con este componente, se utilizó para ahorrar el tiempo de armado del mecanismo móvil.
Armarlo con componentes por separado	.Compuesto por: 2 motores DC, mecanismo de dirección para doblar, y chasis para el soporte de los motores y las ruedas.	Aunque era relativamente más barato, debido a que se contaba con el sistema prefabricado, se ahorró la dificultad de diseñar esta alternativa.

3.4. Comunicación

Opción	Características	Conclusión
WiFi ESP8266	Soporta los buses de comunicación SPI, I2C, UART. Voltaje de trabajo entre 3V y 3.5V. Es bastante económico y pequeño.	Debido a lo económico y fácil de implementar la comunicación se utilizó este dispositivo.
Bluetooth HC-06	a Alimentación de tensión 3.3V. Corto alcance. Sólo puede actuar como esclavo	Tiene un alcance más corto que el dispositivo WiFi.

4. Metodología de diseño utilizada

4.1. Hardware

Para el diseño e implementación del hardware se siguieron una serie de pasos para llegar al objetivo. Entre ellos se encuentran:

1. Buscar sensores que se van a utilizar para el proyecto, así como las hojas de datos para observar los valores de tensión en el que va a funcionar correctamente.
2. Crear el diagrama para ver la simulación del circuito de manera que se observe que los valores de entrada son los correctos y evitar el daño de algún componente del circuito.
3. Armar el circuito para crear un prototipo de manera que se pueda ver el funcionamiento de los distintos sensores, esperando que lo hagan de la mejor manera.
4. Se deben soldar los distintos componentes y probar su funcionamiento.

4.2. Software

El software contempla la parte del que controla los movimientos del carro y la que envía y recibe las señales que genera el robot.

En el caso del software que controla el carro se realizó en Arduino y los pasos para su creación corresponden a:

1. Analizar los movimientos que va a realizar el carro.
2. Observar el uso de cada uno de los sensores para determinar las señales que va recibir el sistema para realizar cada acción.
3. Estudiar cuándo se va a realizar cada uno de estos movimientos para activar las distintas señales de manera que el carro se mueva correctamente.
4. Crear el módulo para la red de manera que se pueda comunicar el código del robot con la aplicación para lograr objetivos para las dos partes.
5. Agregar las señales de entrada y salida que provienen de la aplicación de la computadora para adaptarlas y lograr que se envíen cuando llegan a una estación y se devuelva a la base cuando reciba el comando apropiado desde la aplicación.

Por su parte, la aplicación de computadora es más fácil de implementar por lo que se siguieron los pasos:

1. Determinar las variables que van a corresponder a entrada y salida del sistema.
2. Crear la conexión con el dispositivo de manera que se puedan escribir y recibir mensajes para lograr objetivos.
3. Diseñar la interfaz gráfica de manera que se desplieguen los datos al usuario y permita la ejecución de la acción para el carro para realizar el retorno a la base.

5. Conclusiones y lecciones aprendidas

Conclusiones

Durante todo el proceso del proyecto se llegaron a varias conclusiones dentro de las cuales se encuentran:

- Un microcontrolador es una unidad de procesamiento muy utilizada en sistemas empujados, debido que es posible tener un procesador, memoria y dispositivos de entrada y salida en un solo circuito integrado.
- El consumo de potencia es un parámetro crítico a tomar en consideración cuando se desarrollan sistemas embebidos. Se requiere que consuman poca potencia para poder satisfacer los requerimientos tanto de portabilidad como tiempo en funcionamiento.
- El espacio es una característica que se tiene que tomar en cuenta por el hecho de que en un carro las dimensiones son reducidas y se debe acoplar todo el sistema en ese espacio.
- Por último, el costo debe ser bajo para que sea accesible para las personas que quieran comprar el dispositivo. Para el caso de este proyecto, no es de utilidad que cada uno tenga un costo elevado porque los warehouses contienen grandes cantidades de estos dispositivos y no le sería económico a la empresa adquirir tantos.

Lecciones aprendidas

Por su parte, el proyecto dejó muchas enseñanzas como se muestra a continuación:

- Mantener el orden en el circuito es importante para lograr encontrar de una mejor manera los errores que puedan existir. Cuando los cables se encuentran mezclados uno con otros dificulta la labor de probar y verificar un circuito.
- Manejar todos los componentes como un módulo es útil para lograr un mejor control del sistema. En caso de que exista un error en uno de los módulos es más sencillo probarlo y no tener que pasar por todo el circuito buscando el error. En caso de que uno de los módulos falle, simplemente se puede desconectar y el resto del sistema sirve.
- Otra ventaja de tener el circuito modularizado es que permite la expansión de manera sencilla sin tener que modificar los otros componentes del sistema.
- Estar pendiente de los diferentes componentes del circuito para detectar si alguno se calienta y evitar que alguno se queme. Para impedir que esto ocurra se puede ir incrementando el valor de tensión poco a poco para evitar daños.