

Instituto Tecnológico de Costa Rica	Segundo Proyecto Programado
Escuela de Ingeniería en Computación	<b>ROBOT CON CONTROL</b>
CE-1102 Taller de Programación	Prof: Jeff Schmidt Peralta
I Semestre 2017	Consultas: grupo facebook
18 de mayo de 2017	correo: <a href="mailto:jschmidtcr@gmail.com">jschmidtcr@gmail.com</a>

## 1. Introducción.

El diseño de periféricos para un sistema computacional es una de las áreas de trabajo de un Ingeniero en Computadores. Los microcontroladores son la base para el desarrollo de computación ubicua. Muchos de los sistemas empujados usan microcontroladores como elemento de procesamiento.

## 2. Interfaz hardware-software.

Un robot es una entidad virtual o mecánica artificial, un sistema electrónico y mecánico realiza acciones o movimientos.

Esta etapa del segundo proyecto da un vistazo a los proyectos típicos de ingeniería en computadores los cuales involucran una parte de hardware y otra parte de software. Aplicando teoría de sistemas, no es suficiente con estudiar o desarrollar los componentes en forma independiente sino que también se deben estudiar las interrelaciones. Es así como algunos de los elementos la conexión y relaciones hardware-software se encontrarán en este proyecto.

El diseño de hardware es simple, permite relacionarse con algunos de los componentes que serán objeto de estudio en cursos futuros y se puede contemplar el potencial y posibilidades que presentan, para poder concretar algunos proyectos para después con el fundamento teórico adquirido, poder optimizarlos.

En esta etapa del proyecto se utilizará la tecnología de microcontroladores, se construirá un circuito electrónico sencillo ya diseñado y se va a construir un componente de software para hacer operativo dicho control con un robot virtual, para el cual se pueden reutilizar ponentes de software del proyecto 1 del curso.

El robot virtual que el grupo de estudiantes va a crear (debe definir un nombre y una imagen para SU robot) es una herramienta sencilla que permite el uso del robot.

## 3. Descripción Funcional Programa.

La presentación de la interfaz con el usuario es **totalmente libre** y será un elemento importante dentro de la calificación del proyecto. Se puede utilizar cualquier interfaz gráfica base para el proyecto.

## Modelo de objetos.

El grupo debe definir los atributos y métodos que se requieran y no están descritos en la presente definición. El modelo de objetos enunciado, debe ser claramente documentado.

El programa debe mostrar al robot y las órdenes básicas o comandos, se ejecutarán desde el control remoto.

### 1. Descripción del robot:

El programa va a ser interactivo, se va a ejecutar desde el control remoto, de forma que cuando el usuario prima alguno de los 5 botones del control, el resultado se mostrará en una consola o ventana.



### 2. Comandos permitidos en el control remoto de 5 botones:

`turnRight()` : el robot gira y camina a la derecha  
`turnLeft()` : el robot gira y camina a la izquierda.  
`presentation()` : el robot habla y se presenta.  
`play/stop()` : el robot inicia o finaliza reproducción de música  
`ourbutton()` : el robot hace una función definida por el grupo de trabajo.

Las respuestas básicas pueden darse por medio de consola o ventana. Las acciones que implican movimiento, deben presentarse por medio de alguna animación.

### 3. Consideraciones a realizar.

Entregables:

- Documentación completa del proyecto
- Descripción de la interfaz gráfica y todos sus componentes.
- Código fuente completo, correcto y debidamente documentado mediante comentarios.
- Tarjeta con la implementación del hardware de control (control remoto).

El control será desarrollado usando la tarjeta Arduino UNO basada en un microcontrolador ATMEL MEGA328P con un programa precargado que envía códigos específicos para implementación de botones y recibe órdenes para encendido y apagado de un LED.

Para completar este proyecto se requiere establecer una comunicación serie con el dispositivo de control.

Para ello se debe de instalar la biblioteca pySerial, la cual puede ser bajada para Windows y Linux y que sirve para las versiones 2.x y 3.x de Python, desde el siguiente sitio:

<http://sourceforge.net/projects/pyserial/files/pyserial/2.7/>

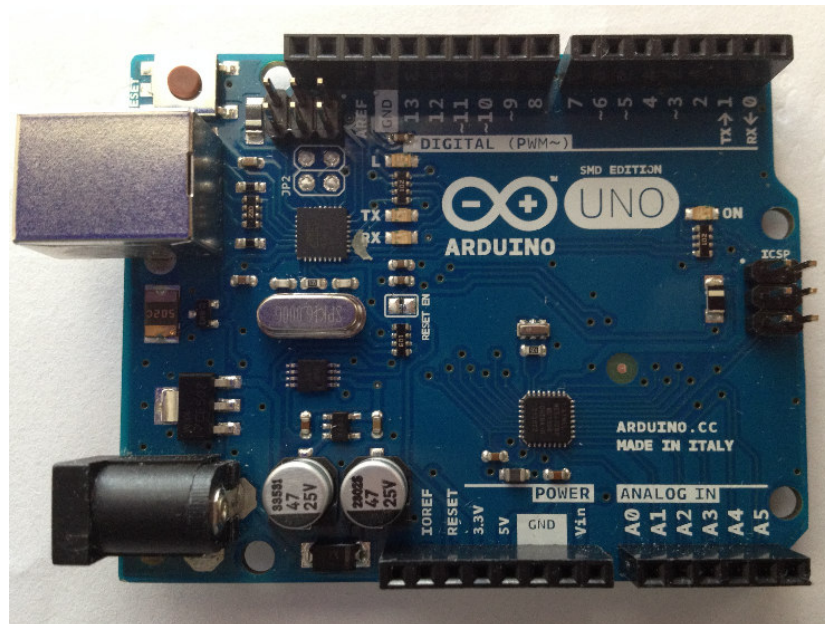
### 4. Diseño de la Tarjeta e interacción con el Arduino.

**Tabla 1.** Comandos para interactuar con el Arduino

Instrucción	Efecto	IN/OUT
'S'	Enciende LED	IN
'N'	Apaga LED	IN
'B1'	Botón B1 presionado	OUT
'B2'	Botón B2 presionado	OUT
'B3'	Botón B3 presionado	OUT
'B4'	Botón B4 presionado	OUT
'B5'	Botón B5 presionado	OUT

**Nota:** *IN* es un dato entrante al Arduino, *OUT* es un dato saliente del Arduino.

En la Figura 1 se muestra una vista superior del Arduino UNO con la numeración de los pines de Entrada/Salida que será usado como base para el diagrama eléctrico de la Figura 2.

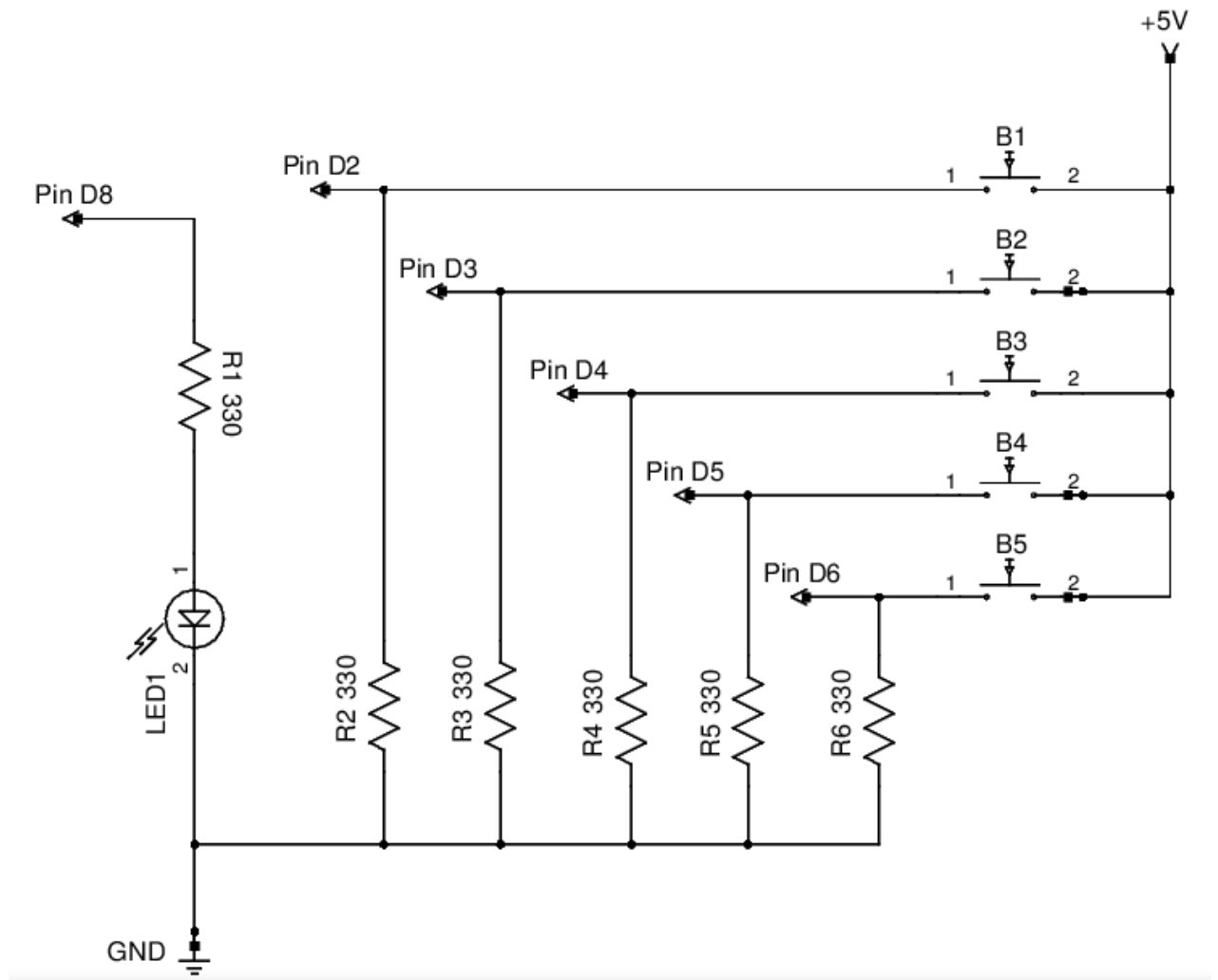


**Figura 1.** Vista superior del Arduino UNO

Los datos enviados y recibidos por el Arduino son datos tipo string codificados en ASCII.

En la Figura 2 se muestra un posible circuito a montar en la tarjeta que contendrá su control de juego para conectar al Arduino. Para la interconexión de las terminales con el Arduino, se pueden usar directamente cables o se recomienda el uso de un conector DB9 (juego hembra-macho).

**Figura 2.** Diseño del circuito electrónico del Proyecto 2.



## 5. Materiales necesarios:

La siguiente es la lista de materiales necesarios para la construcción del circuito:

Placa pre-perforada con lands de cobre sin contactos en la parte de cobre.

5 push buttons normalmente abiertos.

6 Resistencias de 330 ohms.

Un LED pequeño de color amarillo.

Cable UTP Cat5 para la preparación de conexiones con el Arduino.

Un conector DB9 hembra con su carcasa (solo si usa ese tipo de conexión con el Arduino).

Un conector DB9 macho con su carcasa (solo si usa ese tipo de conexión con el Arduino)

#### 4. Documentación.

La documentación interna en el programa fuente, debe contener antes de definir cada función, al menos una explicación de lo que realiza la función, las entradas, salidas y restricciones consideradas.

La documentación externa debe incluir:

- Tabla de contenidos
- Introducción
- Descripción del problema.
- Modelo de objetos: incluyendo diagrama de clases (investigar)
- Dificultades encontradas: problemas en el desarrollo y que se hizo para corregirlos
- Análisis de resultados. (incluyendo corridas de ejemplo)
- Bitácora de actividades: se deben ir anotando todas las actividades, tipo de actividad, su descripción, responsable y duración.
- Estadística de tiempos: un cuadro que muestre un resumen de la Bitácora de Actividades en cuanto las horas **REALES** invertidas. Ejemplo:

<b>FUNCION</b>	<b>Integrante 1</b>	<b>Integrante 2</b>	<b>TOTAL</b>
Análisis de requerimientos	xx horas	xx horas	xx horas
Diseño	xx horas	xx horas	xx horas
Investigación de funciones	xx horas	xx horas	xx horas
Programación	xx horas	xx horas	xx horas
Circuito e interfaz	xx horas	xx horas	xx horas
Documentación interna	xx horas	xx horas	xx horas
Pruebas	xx horas	xx horas	xx horas
Elaboración documento	xx horas	xx horas	xx horas
<b>TOTAL</b>	<b>xx horas</b>	<b>xx horas</b>	<b>xx horas</b>

- Conclusión personal de los integrantes del grupo.

## 5. Evaluación.

Documentación		25%
Interna	5%	
Externa	20%	
Resultados (ejecución, eficiencia, manejo correcto de objetos y estructuras planteadas, presentación del hardware y del software, creatividad)		
Funciones:		
Resultados		75%
Manejo de la interfase	15%	
Modelo de objetos	5%	
Control remoto	30%	
(circuito, funcionalidad)		
Comandos (5 puntos c/u)		
turnRight	5%	
turnLeft	5%	
presentation	5%	
play/stop	5%	
ourbutton	5%	

## 6. Aspectos Administrativos.

- La tarea es **individual** y se debe entregar hasta el día 02/06/2017 a las 9:30 am, en forma electrónica, en un archivo comprimido con el nombre de los estudiantes de grupo, que contenga TODO lo necesario para poder ejecutarla. Ahí se asignará una cita de revisión del programa. **No se aceptarán tareas después de la fecha y hora indicadas.** Debe enviarse un archivo readme.txt con la versión de Python a utilizar para la revisión y alguna otra indicación que se considere importante. Se debe enviar al correo [tareasintrotaller.ce@gmail.com](mailto:tareasintrotaller.ce@gmail.com)
- La defensa o revisión del proyecto es indispensable y deben estar todos los miembros del grupo. En esta revisión se preguntará sobre aspectos relacionados con funcionalidad, así como sobre el código. Los estudiantes deben mostrar TOTAL dominio de estos dos temas, de lo contrario, el proyecto puede ser considerado como una copia.
- En caso de probarse algún tipo de fraude en la elaboración de la tarea se aplicarán todas las medidas indicadas al inicio del curso, incluyendo una carta al expediente del estudiante.
- Se debe incluir en el archivo comprimido la documentación solicitada. Debe entregarse en formato electrónico (archivo .doc .odt o .pdf).
- No se aceptarán tareas cuyo archivo sobrepase 3.0 mb de espacio en disco.
- Cualquier falta a los aspectos aquí enunciados implicará pérdida de puntos.

- **El profesor se reserva el derecho de calificar forma y fondo de las actividades tomando como referencia la mejor actividad presentada**

## **7. Bibliografía.**

Documentación técnica Python y Arduino

## **8. Consultas.**

Puede dirigir cualquier consulta a [jschmidtcr@gmail.com](mailto:jschmidtcr@gmail.com) o al grupo del curso en facebook.