Kevyn Guadamuz Rojas

Carnet: 2017021057

Roger Valderrama

Carnet: 2017113167

Taller de Programación

Tarea Programada # 2

Grupo #1

Profesor: Jeff Schmidt

Fecha de Entrega:  
Viernes 2 de Junio

2017

**Índice:**

**Introducción…………………………………… Pag #3**

**Descripción del Problema…………………….. Pag #4**

**Diagrama de Clases…………………………… Pag #5**

**Dificultades Encontrada……………………… Pag #6**

**Análisis de Resultados………………………... Pag #7**

**Bitácora de Actividades………………………. Pag #9**

**Estadísticas de Tiempo……………………….. Pag #12**

**Conclusión…………………………………….. Pag #13**

**Introducción**

El objetivo de este proyecto es desarrollar la programación orientada a objetos con la combinación de la implementación del hardware para que puedan ser ejecutadas las funciones definidas del objeto por medio de un control remoto ya sea cableado o inalámbrico.

El control remoto se implementará por medio de la placa base Arduino para que esta se comunique con el computador y de esta manera finalmente con Python y la aplicación que hemos creado.

Para lograr esto se requiere investigar sobre conceptos de electrónica, la forma y comando para la programación de la placa Arduino y los módulos necesarios para cumplir que el control remoto se comunique en si con la aplicación de esta manera ampliaremos los conocimientos para la preparación de profesional y lograrse desempeñar en el campo laborar de manera eficiente.

El objetivo de este proyecto es descubrir y desarrollar por medio de la practica la técnica Programación Orientada a Objetos, consiste en que programar, en el lenguaje Python, un juego con fines solamente de diversión que deberá realizar diferentes funciones las cuales deberán ser presentadas por medio de animaciones.

Se trabajará conceptos como soladura, circuitos desarrollando la parte de electrónica que es parte fundamental de la carrera Ingeniera en Computadores.

**Descripción del Problema**

El objetivo principal es hacer un programa interactivo, este se va a ejecutar desde el control remoto, de forma que cuando el usuario oprima alguno de los cinco botones del control, cada botón realiza una función en específico el resultado se mostrará en una consola o ventana.

Los comandos que deberá realizar el interactivo serán los siguientes:

* 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.

El robot deberá ser implementado en programación orientada a objetos. Se deberá definir los métodos y atributos de cada objeto y no esta descritos en la presente definición. El modelo de objetos enunciado, debe ser claramente documentado.

Se deberá investigar la forma de la programación en Arduino y como lograr la comunicación entre la placa Arduino y el lenguaje Python.

Se deberá implementar un circuito que comunique los botones con el Arduino, soldando cada parte para que cada botón envié una señal y se ejecute una función de la aplicación. Poniendo en práctica la motora fina del estudiante y la capacidad para soldar y trabajar con la parte de electrónica que es parte crucial de la carrera Ingeniería en Computadores.

Diagrama de Clases:

|  |
| --- |
| Clase MyThread |
| Atributos:   * MySerial: Puerto comunicación * Queque: Cola |
| Métodos:   * Run(): Lee el mensaje de Arduino y convierte el mensaje en otro string para llamar a otra función que lo interpreta y llama a la función correspondiente. |

|  |
| --- |
| Clase Robot |
| Métodos:   * \_\_init\_\_(): Inicializa el puerto Pyserial para la comunicación con el arduino. |

|  |
| --- |
| Clase GUI |
| Atributos:   * Master: La ventana principal * Myserial: Inicia la función para que se lea constante la comunicación con el Arduino. |
| Métodos:   * cargarImagen(): Cargas las imágenes necesarias para el desarrollo de la aplicación * updateMe(): Verifica el dato que se recibe de Arduino y llama a la función correspondiente. * Right(): Animación que mueve el robot hacia la derecha * Left(): Animación que mueve el robot hacia la izquierda * Own(): Animación definida por los programadores * Play(): Reproduce la música * Pause(): Pausa la música * Presentation(): Animación de la presentación del robot. |

**Dificultades Encontradas**

**Sincronización Arduino y Tkinter:**

Uno de los principales problemas encontrados en la realización del proyecto fue lograr la comunicación del Arduino con Python. Se implementó la librería Pyserial, que es la librería por defecto para que el interfaz de Python reciba los datos provenientes de un puerto serial.

Se implementa un Thread propio para que este se ejecute al mismo tiempo de la ventana de la interfaz y que no haya ningún choque de información ni de prioridad, ya que el problema que persistía era que no se lograba mediante un hilo normal de Python debido a que según nuestro conocimiento al momento que la ventana es principal no permite ejecutar el ciclo infinito para que se lea constantemente la información del Arduino. Por lo que se implementó un Thread propio para que realizara esta función.

**Como leer el dato del Arduino:**

Se investiga como asignarle una variable a los datos que este enviando el Arduino para que en Python puedan ser leídos correctamente. Para la interpretación de los datos que se reciben de Arduino se implementa una función que lea lo que el Arduino envía hacia el equipo y ejecute la función mediante el dato que se reciba.

Esta función “MyUpdate” se ejecuta con junto a la ventana, gracias a la implementación del Thread propio que se utilizó para que este siempre leyendo constantemente los datos que se reciben del Arduino y de esta forma a cualquier instante que un botón sea presionado la aplicación ejecuta la animación correspondiente a ese botón

**Reproducir audio:**

Para la reproducción de audio necesitábamos implementar un Thread para que el audio se reprodujera mientras se ejecutaba la ventana y así al momento de apretar el mismo botón este se detuviera.

Debidos a los problemas que se han tenido con los Threads, el audio no pudo ser implementando con los Threads pero preguntado y nos sugirieron usar la librería PyGame para ejecutar el audio, en efecto se utiliza la librería dicha anteriormente y reproduce los audios sin ningún problema en cuanto al reproducirlos juntamente cuando se ejecuta la ventana.

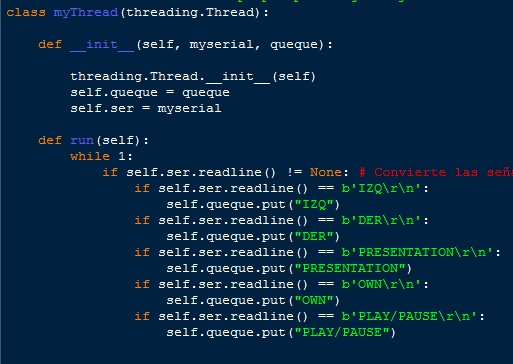
**Soldar el circuito en la Tarjeta Perforada:**

Debido a la inexperiencia con el trabajo con circuitos, la parte de soldar el circuito a la tarjeta perforada nos complicó un poco el desarrollo del proyecto, sin embargo, con la ayuda de la investigación se logra un resultado muy bueno con respecto a la soldadura.

**Análisis de Resultados**

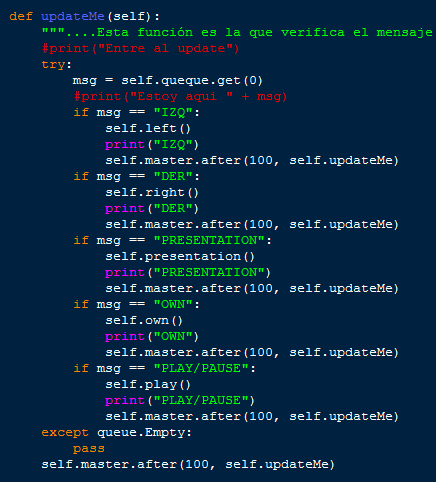
**Sincronización de Arduino y Tkinter:**

Se implementa el Thread propio y este funciono de la manera correcta, gracias a esta implementación se soluciona el problema de ejecutar la ventana y leer los datos del Arduino al mismo tiempo.



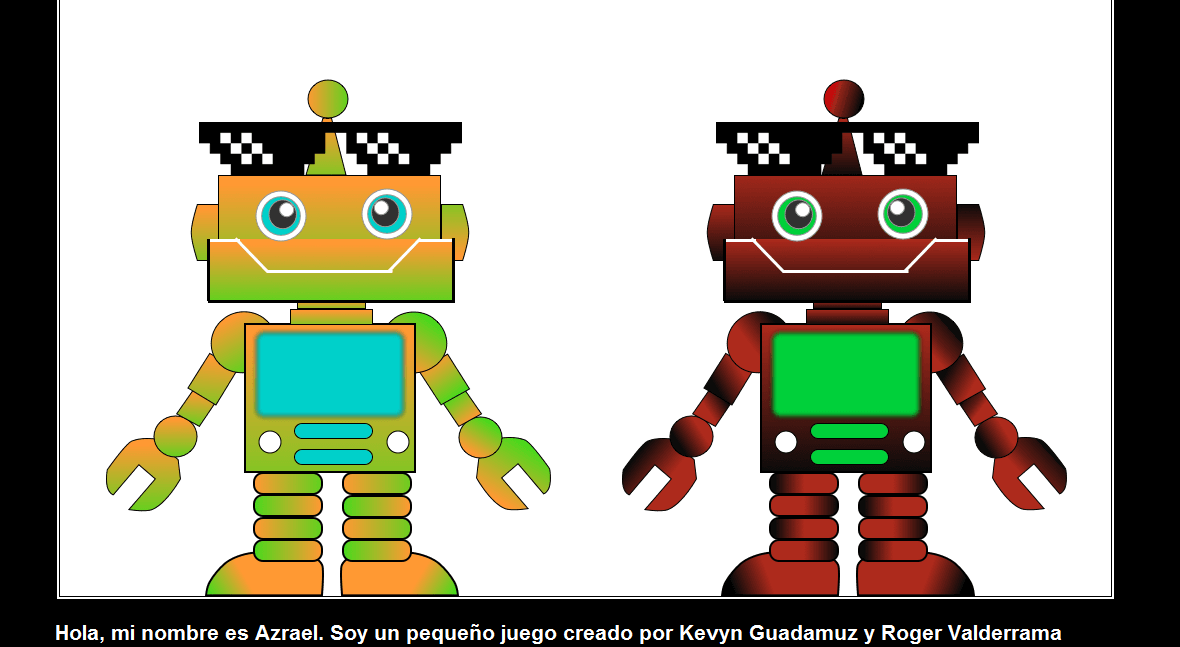
**Leer el dato del Arduino:**

Para leer Arduino se implemente una función que interpreta las señales o comandos que son enviados desde el Arduino y según sea el comando recibido este realiza la función definida.



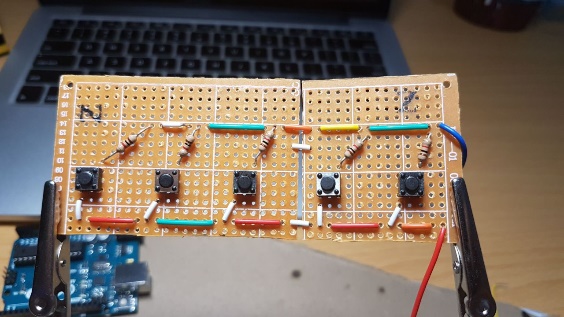
**Reproducir audio:**

Para la reproducción del audio se implementa la librería PyGame permitiendo así que ejecuten PyGame y Tkinter al mismo instante. Tambien en temas de utilización de recursos, PyGame permite ahorrar mucho espacio en el disco duro donde este es almacenado

**Soldar el circuito:**

Debido a la inexperiencia, como se comentó anteriormente, se complicó la parte de soldadura en la tarjeta. Pero con la ayuda de diferentes fuentes de información logramos soldar el circuito correctamente.

**Bitácora de Actividades**

Sábado 20 de mayo: Se investiga sobre módulos Bluetooth y como se implementan con Arduino. Roger Valderrama (6 horas)

Domingo 21 de mayo: Se investiga sobre comandos AT y se empieza con el diseño de los frames para las animaciones en Sketch. Roger Valderrama (6 horas)

Lunes 22 de mayo: Implementación de POO.  
Se inicia con la implementación de la interfaz gráfica en Programación Orientada a Objetos. Kevyn Guadamuz Rojas (5 horas)

Miércoles 24 de mayo:

Instalación Pyserial.  
Se instala Pyserial en el equipo para empezar a desarrollar el código y la forma de cómo comunicar Arduino con Python. Kevyn Guadamuz Rojas (2 horas).

Se crea el repositorio en Git y se desarrolla código principalmente de correcciones. Roger Valderrama (3 horas)

Comunicación Arduino con Python.  
Se inicia el código para la comunicación del Arduino con Python.  
Kevyn Guadamuz Rojas (3 horas).

Pruebas con botones.  
Se realizan pruebas para determinar si al presionar un botón realiza una función cualquiera, pero no funciona por el momento. Kevyn Guadamuz Rojas (1 hora).

Estructura de Objetos.

Se restructura el código con un modelo orientado a objetos y se agrega la función cargarImagen. Roger Valderrama (1 hora)

Viernes 26 de mayo:

Circuito con los 4 botones.  
Se arma digitalmente el circuito para luego pasarlo en físico en la Prothoboard y realizar las pruebas con los demás botones. Kevyn Guadamuz Rojas (30 minutos).

Programación del Arduino.  
Se realiza la programación del Arduino, se le asigna los pines correspondientes a cada botón. (1 hora).

Interpretar datos de Arduino en Python.  
Se intenta implementar la forma en que Python interprete los datos provenientes del Arduino y se logra por medio de strings. Kevyn Guadamuz Rojas (3 horas).

Domingo 28 de mayo:

Se investiga y finalizan los frames para las animaciones Left y Right. Roger Valderrama (8 horas).

Lunes 29 de mayo:

Rediseño del circuito.  
Se rediseña el circuito para agregar un quinto botón. Kevyn Guadamuz y Roger Valderrama (30 minutos).

Documentación.  
Se inicia con la documentación del proyecto, interna y externa. Kevyn Guadamuz y Roger Valderrama (3 horas)

Pruebas Interfaz y Botones.  
Se Realizan las pruebas para implementar las animaciones que se ejecuten desde los botones, pero no funciona. Kevyn Guadamuz y Roger Valderrama (5 horas).

Creación de Thread.

Se investiga sobre como corre nuestra interfaz y la necesidad de tener un hilo conectado al Arduino y se toma la decisión de definir un Thread entonces se investiga sobre queue, y así poder leer las señales del Arduino y también mostrar alguna animación. Roger Valderrama (7 horas)

Martes 30 de mayo:

Implementación de Animaciones.  
Se empieza a crear los algoritmos de las animaciones que va realizar la aplicación. Kevyn Guadamuz Rojas (5 horas)

Se comprimen los frames para las animaciones. Roger Valderrama (1 hora)

Miércoles 31 de mayo:

Impresión de la caja. Kevyn Guadamuz y Roger Valderrama (4 horas)

Se inicia con el diseño del circuito para soldar en PCB y se decide diseñar otra caja con cortes diferentes para cortar el día siguiente. Roger Valderrama (5 horas)

Jueves 1 de mayo:

Añadir Frames.  
Se añaden los frames de las animaciones y se prueban que funcionen con normalidad. Kevyn Guadamuz Rojas (4 horas)

Se implementa la música.  
Se implementa la reproducción de audio en el proyecto. Kevyn Guadamuz (1 hora)

Se finaliza la documentación  
Se finaliza la documentación interna y externa de la Tarea Programada. Kevyn Guadamuz (5 horas)

Reparando detalles en las animaciones.

Se trabaja en los detalles de la aplicación como las etiquetas, centrar imágenes y muchas cosas más (3 horas)

Se imprime la nueva caja. Roger Valderrama (2 horas)

Se termina de soldar el circuito. Roger Valderrama (8 horas).

Se arma la caja. Roger Valderrama (1 hora)

Se contribuye a la documentación externa. Roger Valderrama (1 hora)

**Estadística de Tiempos**

|  |  |  |  |
| --- | --- | --- | --- |
| **FUNCION** | **Kevyn Guadamuz** | **Roger Valderrama** | **TOTAL** |
| Análisis de requerimientos | 13 horas | 15 horas | 28 horas |
| Diseño | 10 horas | 20 horas | 30 horas |
| Investigación de funciones | 5 horas | 14 horas | 21 horas |
| Programación | 20 horas | 14 horas | 34 horas |
| Circuito e interfaz | 21 horas | 14 horas | 35 horas |
| Documentación interna | 2 horas | 1 hora | 2 horas |
| Pruebas | 2 horas | 1 hora | 3 horas |
| Elaboración documento | 5 horas | 1 hora | 6 horas |
| **TOTAL** | **78 horas**  **\*\*Nota: Se hace demasiado debido a que algunas horas están repetidas e otros aspectos \*\*\*** | **80 horas** | **158 horas** |

**Conclusión**

Kevyn Guadamuz:

Con este proyecto aprendí mucho sobre las animaciones, conocí más Tkinter. Amplié mi conocimiento en el desarrollo de circuitos y la programación de un Arduino, antes de este proyecto, difícilmente sabía que era y gracias a este proyecto se logró trabajar con uno.

En el trascurso del proyecto aparecieron retos que se lograron superar de una manera correcta, siempre con la mentalidad de aprender para así lograr en un futuro desempeñarse de la mejor manera y aprovechar todo ese conocimiento que estamos obteniendo hoy y ponerlo en práctica en nuestros lugares de trabajo.

Se conoció más de la parte de electrónica combinada con la parte de la informática. El desarrollo de software y hardware son las bases de nuestra carrera y por proyectos como este son puestas en prácticas las dos facetas. La programación y la construcción de circuitos.

Roger Valderrama:

Con este proyecto se amplió el conocimiento de cómo funcionan los programas por atrás en sentido de Threads, y como ciertos procesos requieren de prioridad. Se amplió todavía más el conocimiento de animaciones ya que tuve que investigar como hace que un personaje camine de perfil para las animaciones y también aprendí mucho de diseño en corte laser.

Finalmente, enfrentamos retos con ámbito vocacional sobre la complejidad de la comunicación hardware-software que nos dan una idea más clara del futuro en esta carrera y en mi caso me motiva a continuar.