

Instituto Tecnológico de Costa Rica

Lenguajes, compiladores e intérpretes

**Proyecto Final - BrailleRead
Documentación externa**

Subgrupo D

Integrantes:

Noel Castro Acosta 2019044736

Daniel Cob Beirute 2021084824

Kevin Lobo Juárez 2020087823

Eder Vega Suazo 2021067844

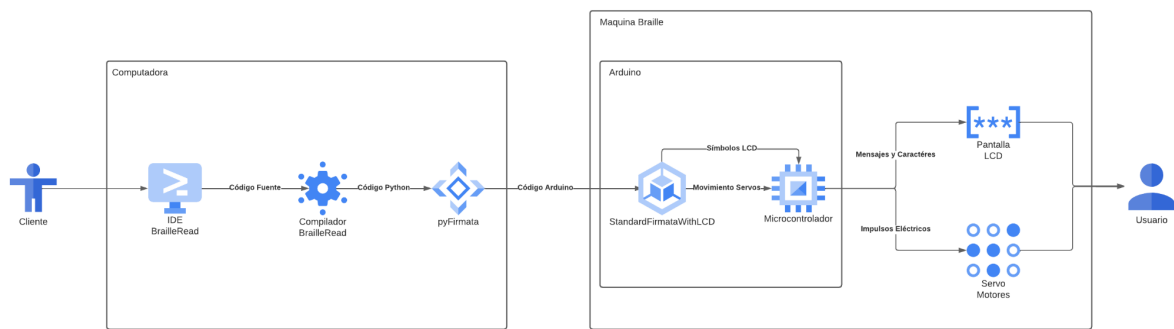
Profesor:

Marco Hernandez Vasquez

Grupo 02

I semestre 2023

Diagrama de arquitectura de solución



Problemas conocidos

El hardware de la máquina braille está construido dentro de un contenedor prototipo hecho con cartón de presentación ya que facilitó el proceso de fabricación del dispositivo. Sin embargo esto causa problemas principalmente respecto a la rigidez del prototipo. Se encontró que si se hace mucha presión sobre los puntos de braille los servomotores que los levantan se pueden despegar de manera que es necesario abrir al prototipo para arreglarlo y hacer que vuelva a funcionar. Además el prototipo es difícil de abrir lo que dificulta aún más la situación.

Cerca de la entrega del proyecto se presentó un problema que no se había observado, ya que es bastante particular. Al usar la función signal sobre un solo servo una vez en todo un programa, no genera la señal esperada en el arduino. Es decir, cuando se presenta esta situación no hay ninguna respuesta de parte del servo para levantar o bajar el punto. Sin embargo, esta situación no afecta en gran medida el proyecto ya que para leer una sola palabra en braille es necesario mover varios servos al mismo tiempo.

En el caso de la interfaz gráfica, presenta un pequeño delay con respecto a las líneas de código y el contenedor donde se puede visualizar el código, por lo que la experiencia puede resultar confusa para el usuario en ocasiones.

Por otra parte, cuando el código muestra un error indica en qué línea es pero no coincide dónde está el error exactamente, por lo que buscar el error por línea queda obsoleto. Sin embargo, funciona como una guía en donde se podría encontrar el error al realizar saltos de línea y verificar que el error lo muestre en otra línea de código.

Al momento de un **Case** se hace imposible la comparación para poder continuar, sino que a pesar de que el valor a comparar con la variable no se cumple está en cada uno de los casos.

Problemas encontrados

La pantalla LCD utilizada en el proyecto cuenta con un controlador I2C que reduce la conexión de la pantalla al arduino a tan solo 4 cables lo que permite dejar libres el resto de pines para los servomotores. Sin embargo *pyFirmata*, la librería utilizada para comunicarle al arduino instrucciones desde *Python*, no cuenta con una manera sencilla de utilizar dispositivos I2C lo cual nos hizo replantear la solución propuesta. La solución final consiste en una modificación al programa firmata que se almacena en el arduino que además de recibir las instrucciones desde python, también controla por sí sólo la pantalla indicando cuál símbolo está siendo representado en braille a través de las posiciones de los servos. Esto permitió tener una versión más eficiente y sencilla de usar del arduino desde el programa de python que únicamente le indicará al arduino cuando debe subir o bajar los servos.

Conclusiones

- El proyecto de crear un compilador propio utilizando Python con Lex-Yacc y la implementación de un dispositivo hardware creado a partir de un Arduino para simular la escritura en braille fue realizado exitosamente
- Se logró desarrollar un lenguaje de programación que permite compilar diferentes operaciones solicitadas.
- La implementación de los servomotores y los pines para simular el código braille en el dispositivo hardware funcione correctamente

Recomendaciones

- Continuar explorando y mejorando el lenguaje de programación desarrollado, agregando nuevas funcionalidades y optimizando su rendimiento.
- Realizar pruebas exhaustivas del dispositivo hardware para asegurarse de su correcto funcionamiento y durabilidad a largo plazo.
- Considerar la posibilidad de expandir el proyecto para incluir la opción de recibir texto a través de diferentes medios, como un teclado o una interfaz de usuario en un dispositivo móvil.
- Explorar oportunidades de colaboración con instituciones o organizaciones que trabajan en el ámbito de la accesibilidad para

personas con discapacidad visual, con el fin de obtener retroalimentación específica sobre la utilidad y la mejora del dispositivo en entornos reales.

- Mantenerse actualizado con los avances tecnológicos en el campo de la accesibilidad y la escritura en braille, para seguir mejorando y adaptando el proyecto en el futuro.

Bibliografía

Beazley.D (S.F). PLY (Python Lex-Yacc). dabeaz-com.Recuperado el 27 de mayo 2023 de https://www.dabeaz-com.translate.goog/ply/ply.html?_x_tr_sl=auto&_x_tr_tl=es&_x_tr_pto=wapp&_x_tr_hl=uk

Llamas, L (2016).Controlar Arduino con Python y la librería PySerial. Luis Llamas Ingeniería, informática y diseño. Recuperado el 12 de junio de 2023 de <https://www.luisllamas.es/controlar-arduino-con-python-y-la-libreria-pyserial/>

Kazarinoff, P. (2021). Using Python and an Arduino to Read a Sensor. Python for Undergraduate Engineers. Recuperado el 12 de junio de 2023 de <https://pythonforundergradengineers.com/python-arduino-potentiometer.html>

Comisión braille española.(2005). Signografía básica. Recuperado el 12 de junio de 2023 de <http://sid.usal.es/idocs/F8/FDO12069/signografiabasica.pdf>

The PyFirmata Developers. (S.F). PyFirmata Documentation. Recuperado de <https://pyfirmata.readthedocs.io/en/latest/>

Python Software Foundation. (2023). Python Documentation. Recuperado de <https://docs.python.org/3/>