



# Lenguaje Arduino

## Instituto Tecnológico de Costa Rica

Daniel Alvarado Chou (201235858), Andre Solis Barrantes(2013389035) and Karen Lepiz Chacon(201300534)

Ingeniería en Computación

Compiladores e Interpretes

Profesor: Esteban Arias Mendez

Sede: Cartago

Semestre II

30 de noviembre de 2016

## Resumen

This document will inform you of the tools used to create a compiler for the Arduino language. Where the grammar will be translated into Spanish and will present slight changes, these will be presented later. Also, this content the grammar BNF used, as well as examples of the functionality of the new version of grammar of the language. It will detail the tools that were used for the execution of the project, and will show important definitions that the reader should handle for a correct understanding of the work

## 1. Introduccion

En este proyecto se quiere crear un compilador en donde se tiene como una base la gramática en el que se quiere trabajar, que, en este caso, corresponde a la gramática de Arduino. La gramática del lenguaje de programación Arduino será traducida al español, en donde no solo se le va a realizar el parser, scanner, sino también va a ser capaz de manejar errores léxicos, sintácticos y semánticos. Se va a presentar el diseño del lenguaje traducido al español, el alfabeto y delimitadores que se utilizaron, así como la explicación de las causas de los errores léxicos, sintácticos y semánticos que se pueden presentar en el lenguaje, y también la manera en el que el programa se recupera del mismo.

## 2. Descripción del Lenguaje Arduino

Es una plataforma electrónica abierta en la que se basa en prototipos en software y hardware libre, que son muy sencillos de entender e implementar, la cual fue creada por Arduino LLC. Esta diseñada para introducir la programación a artistas y nuevas personas que no están familiarizadas con el desarrollo del software. El lenguaje es capaz de soportar los lenguajes C y C++.

Características.

- Librería Wiring para entradas y salidas.
- Basados en dos partes Setup y Loop.
- Es código libre.
- Tipo Compilado.
- Es compatible con los sistemas operativos : Linux, Windows

## 3. Diseño

- La base para la gramática que se va a utilizar en el proyecto será la gramática original del Lenguaje
- La nueva gramática será traducida en el idioma español, es decir para escribir un While se pondrá MIENTRAS.
- Las operaciones de decremento(−) e incremento(++), originalmente se escribiría variable++ o variable−, pero en la nueva gramática será: ++variable y −variable. La razón de este cambio es que es más fácil de analizar semánticamente.
- Las funciones principales Loop() y Setup() no se podrán reescribir como se puede hacer en la gramática

tica original, esto con el motivo de que sean funcione ya predeterminadas del lenguaje.

- Al escribir la gramatica se debe tener cuidado de la ambigüedad.

- Las palabras reservadas y nombres de los no terminales de las que estar compuesto la gramatica deben ser palabras significativas, esto con el motivo que si alguna otra persona quisiera tener el acceso a la gramatica su legibilidad sea mas facil.

## 4. Herramientas Usadas

Las herramientas que se van a usar para la generacion del parser y scanner seran JFlex y Cup. Las razones por las que se eligieron estas herramientas se presentara a continuacion:

JFlex

- Permite generar analizadores de manera rapida.
- La sintaxis es fcil de comprender e implementar.
- Es independiente de la plataforma.
- Integracion con Cup.

Cup

- Declaracion de simbolos no terminales y terminales.
- Declaracion d precedencias.
- Se define un simbolo inicial de la gramatica.
- Defincin de relgas de produccion.

La instalacion de estas herramientas son bastante sencillas, basta con descargar el .zip de JFlex que esta presente en la pagina oficial, e importar esas librerias al proyecto de Netbeans (en este caso), y ya se podra compilar archivos .flex y .cup.

## 5. Conclusiones y Obesarvaciones

El desarrollo de un compilador para un lenguaje es una tarea bastante tediosa, ya que todo depende del buen diseo de la gramtica. Si la gramtica no est bien construida o diseada, ya sea por errores e ambigüedad o por escritura, el resto del proceso puede resultar defectuoso o incluso se tenga que volver a crear. Se tiene que tener bien claro la manera en la que se quiere que el lenguaje funcione y como interpretar las sentencias que se escriban en el mismo.

La buena comprension sobre el funcionamiento de un compilador, asi como la tarea que realiza cada parte, as como comprender la manera en la que funciona el lenguaje en el que se va a trabajar, ya sea de

manera sintctica y de manera semntica. Entender el lenguaje facilitara ligeramente la construccin de la gramtica que se desea usar para el compilador.

Algunas de las observaciones presentes durante la elaboracin del proyecto se encuentran las siguientes:

- Las herramientas que se elijan utilizar puede afectar de manera positiva o negativa en el proyecto, en este caso se utilizaron JFlex para el scanner y Cup para el parser, las razones por las cuales se eligieron se encuentran en la seccin de Justificacin de este mismo documento.

- Encontrar informacin sobre como realizar los analizadores y entender la manera que funciona, para poder crear nuestra propia lgica fue difcil, afectando la dificultad de la creacin del compilador.

- El lenguaje Arduino en particular, se tuvo que crear una propia gramatica en BNF, puesto que no estaba publicada de manera pblica en ningn repositorio (git) o en la pgina oficial del lenguaje

## Referencias

- [1] Capitulo 5.Analsis Semntico. (s.f.). Recuperado el 23 de Noviembre de 2016, de <http://arantxa.ii.uam.es/alfonsec/docs/compila5.htm>.
- [2] Castro, R. A. (Octubre de 2008). Integracion de JFlex y Cup. Recuperado el 23 de Noviembre de 2016, de <http://www.rafaelvega.info/wp-content/uploads/Articulo.pdf>
- [3] Definicion:Arduino. (5 de octubre de 2012). Recuperado el 23 de Noviembre de 2016, de Arduino: <http://jamangandi2012.blogspot.com/2012/10/ques-arduino-te-lo-mostramos-en-un.html>
- [4] Vigilante. (10 de Diciembre de 2007). Analizador lxico, sintctico y semntico con JFlex y CUP. Recuperado el 23 de Noviembre de 2016, de CRY-SOL: <http://crysol.org/es/node/819>