

Proyecto: Carro controlado vía Bluetooth con lenguaje propio.

Integrantes:

- Cruz Arredondo César
- Figueroa Luis
- López Ayala Eric Alejandro
- Olea Zúñiga Jonathan

Recursos a ocupar:

- *Lenguajes de programación*
 - Java
 - Python
 - C
- *Hardware*
 - Placa Arduino Mega o similar
 - Carro con dos motores
 - Sensor de Temperatura LM35
 - Módulo Bluetooth HC-05
 - LED'
 - Cámara con ranura SD
 - Buzze

Resumen

Mediante una interfaz gráfica, el usuario podrá darle indicaciones al robot para moverse o realizar acciones específicas [véase *Anexo I*]. Los datos ingresados serán validados para verificar si los datos ingresados por el usuario corresponden al lenguaje (léxico). La comunicación será mediante Bluetooth.

Descripción Detallada

Físico (Robot)

- El movimiento del robot será en cuatro direcciones: adelante, atrás, derecha e izquierda.
- Tendrá control de velocidad.
- Contará con LED's, los cuales variarán su intensidad.
- Contará con una cámara, la cual tomará foto y video. Dicho material multimedia será almacenado en una tarjeta SD que el usuario ingresará a la cámara.
- Tendrá un sensor de temperatura, cuya lectura será enviada a la computadora con el fin de modificar sentencias de control.

Java

- Tendrá el analizador léxico en BYACC el cual generará las instrucciones para el robot enviadas a través de un socket para el robot. *Anexo I*. Tal analizador aceptará secuencias de instrucción simples, las cuales podrán estar asignadas a variables que el usuario defina, para su posterior invocación. Además, será posible manejar ciclos

como *while* y *for* (incluyendo variables de tipo entero y la que se lea por el sensor de temperatura ya definida como TEMP).

Python

- Mediante el uso de sockets de flujo, recibirá del programa en Java los comandos que se enviarán al Arduino vía Bluetooth.
- Reenviará los datos que conciernen al programa en Java, tal como la temperatura.

Arduino

- Se recibirán códigos comandos, los cuales se identificarán para la ejecución de la acción correspondiente.
- Enviará una respuesta al término de cada una de sus acciones realizadas. Sí la acción realizada fue la medición de temperatura en dicho momento, se enviará este dato. En otras acciones bastará con un 'recibo'.
- Se usará PWM para el control de velocidades de motor.

ANEXO I. ACCIONES DEL ROBOT

A continuación, se enlistan las acciones que podrá realizar el robot.

- Moverse hacia adelante, atrás, izquierda y derecha.
- Aumento y disminución de velocidad.
- Prendido, apagado y control de intensidad de luces.
- Captura de fotografías y toma de video.

ANEXO II. GRAMÁTICA DEL LENGUAJE PARA CONTROL DEL ROBOT

```
%token BLTIN
%token IF ELSE WHILE FOR PRINT
%token FUNC PROC RETURN
%token EQ NEQ GT GE LT LE
%token FUNCTION PROCEDURE
%token ARG
%token NUMBER
%token VAR
%token VARCRD
%token TEMP
/* Operadores */
%right '='
%left '-' '+'
%left '*' '/' '^'

%%
/* Reglas gramaticales */

list:
| list '\n'
| list asgn '\n' {} //asignacion a variable tipo double
```

```

| list stmt '\n' {} //statement, condicionales o decisiones
| list exp '\n' {} //expresión de tipo double
| list asigncoord '\n' {} //asignacion a variable tipo coordenada
| list expcoord '\n' {} //expresión de tipo coordenada
;

asgn: VAR '=' exp {}
;
/*Statements*/
stmt: exp {}
| PRINT exp {}
| asigncoord { System.out.println("nabla");
maq.banderaVar=true;
maq.code("printVar");
maq.code("STOP");
}
| expcoord {}
| while cond stmt end {}
| if cond stmt end {}
| if cond stmt end ELSE stmt end {}
| for '(' inicio ';' cond ';' inc ')' stmt end {}
| '{' stmtlist '}' {}
;
cond: '(' exp ')' {}
;
while: WHILE {}
;
if: IF {}
;
for: FOR {}
;
inicio: exp {}
;
inc: exp {}
;
end: {}
;
stmtlist: {}/*nada*/
| stmtlist '\n'
| stmtlist stmt
;

exp: NUMBER {}
| VAR {}
| TEMP {} /*variable física del sensor de temperatura*/
| asgn {}
| BLTIN '(' exp ')' {}/*buitlin para sin cos log*/

```

```

| exp '+' exp          { maq.code("add"); }
| exp '-' exp          { maq.code("sub"); }
| exp '*' exp          { maq.code("mult"); }
| exp '/' exp          { maq.code("div"); }
| exp '^' exp          { maq.code("powN"); }
| '(' exp ')'          { $$ = $2; }
/* Operaciones condicionales/booleanas */
| exp EQ exp           { maq.code("eq"); }
| exp NEQ exp          { maq.code("neq"); }
| exp GT exp           { maq.code("gt"); }
| exp GE exp           { maq.code("ge"); }
| exp LT exp           { maq.code("lt"); }
| exp LE exp           { maq.code("le"); }
;

/*gramática de instrucciones al robot*/

asigncoord:  VARCRD '=' binstr  {}
;

binstr: binstr instr  { maq.code("suma"); }
| 'c'                {}
| expcoord            {}
;

expcoord:  expcoord '+' expcoord {}
| expcoord '-' expcoord {}
| sec      {}
;

sec:  'c'      {}
| sec instr  {}
| VARCRD      {}
;

instr:  'a'      { maq.code("avanza"); }
| 'r'      { maq.code("retrocede"); }
| 'i'      { maq.code("izquierda"); }
| 'd'      { maq.code("derecha"); }
;

%%

```

Ejemplo de programas:

```

>>crda= c v a a d f a a
>>crdb= c d f a a d f a a
>>crdc= c u a a a a d
>>crde= crda + crdb + crdc
>>for(i=0; (i<5); i=i+1) { if((( i / 2)*2)== ind) { c v + crde}
else {c l + crde}}

```

Avanza formando cinco cuadrados separados entre si, de forma rápida si el número de cuadrado es par y lenta si es impar, tomando en cada vértice de ellos una foto. Cuando acaba de formar un cuadrado, prende la luz led y la mantiene prendida hasta que dibuja el siguiente cuadrado.

```
>>crda= c l IV a d a i a
```

```
>>crdb= c v DV a i a d a
```

```
>>for(i=0; i<10; i=i+1) {if(TEMP<=25) {crdba} else {crdb}}
```

Avanza formando una escalera hacia la derecha si la temperatura es menor a 25°C o hacia la izquierda si es mayor a 25°C.