

Análisis de software para robots terrestres

Franco D. Bernstein ^{a,*}; Juan Ramon Viera ^{*}

^a Facultad de Ingeniería, Universidad Nacional de Misiones (FI-UNaM), Oberá, Misiones, Argentina.

^b GID-IE, FI-UNaM, Juan Manuel de Rosas 325, Oberá, Misiones, Argentina.

e-mails: bernstein593@gmail.com, juanvier77.com@gmail.com.

Resumen

En el presente trabajo se abordara temáticas como el análisis del software de un robot en particular el cual es un N6, características físicas también se interioriza junto a el lenguaje python el cual se hace un análisis general del mismo y se aplica este lenguaje al robot N6 con el fin de simplificar la programación aplicada al sistema, finalmente se procede a describir procedimientos a llevar a cabo para hacer la correcta interpretación de este lenguaje al sistema en cuestión.

Palabras Clave – *Firmware, N6, Python, Robot, Software, Xbee*

1. Introducción

El presente trabajo se centra en el análisis del robot N6 desarrollado por la empresa RobotGroup, el cual su desarrollo se enfoca en poder llevar a las escuelas una herramienta didáctica de fácil uso para que todos puedan introducirse en la programación de forma didáctica por medio de MiniBloq, una herramienta simple de interpretar, ideal para personas que se introducen en la programación.

El trabajo se basa en el estudio de las características de dicho robot, y en como modificar el firmware para que el mismo pueda ser interpretado por medio del lenguaje de programación Python, herramienta de programación de simple comprensión que conlleva grandes ventajas en comparación con otras.

Finalmente se presentan script que se toman como ejemplos de programación por medio de Python en el cual el robot N6 las toma y lleva a cabo las determinadas tareas impuestas en el código desarrollado.

2. Descripción del Robot N6

El robot está compuesto por su unidad central de procesamiento (CPU) el cual a su vez se deriva en diversas ramas como ser, sensores, motores, puertos, etc. A continuación se presenta un diagrama de bloques que representa de manera global la composición del robot en cuestión.

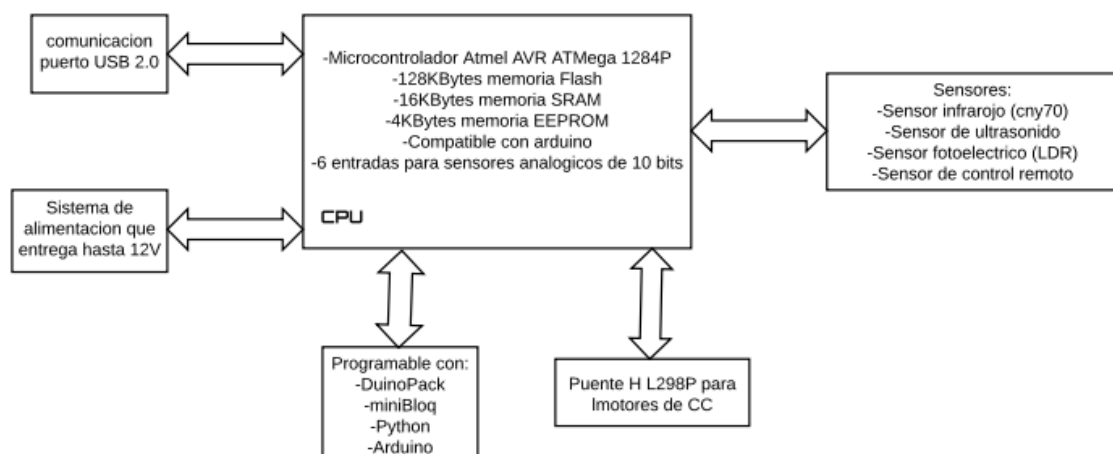


Figura 1: Diagrama de bloques robot N6

EL robot se puede dividir en 3 grandes bloques los cuales son, sensores, motores, y unidad central de procesamiento (CPU), estas se describen a continuación junto con otras características del mismo.

- **Sensores:**

Este robot está compuesto por 2 sensores cny70, 1 sensor ultrasónico, 2 sensores LDR, 1 sensor de control remoto, 1 shield de expansión que permite agregar otros sensores que requiera el usuario.

- **Puente H L298P**

Este controlador hace que sea mas sencillo el poder controlar tanto motores CC o inclusive motores paso a paso, tiene la capacidad de controlar motores de hasta 12 V con una corriente límite de 2A.

- **Comunicación**

El dispositivo puede comunicarse a través del puerto USB 2.0, es compatible con Arduino, también posee 6 entradas para dispositivos de tipo analógico (10 bits) en donde cada uno de estos admite una corriente de 40 mA (200mA como máximo) si es configurado como salida digital.

Una vez que se identifica las partes del robot se procede al desarrollo el epicentro del proyecto el cual consiste en la instalación del firmware que nos permitirá realizar la comunicación a través del lenguaje propuesto.

Instalación del firmware

Los pasos a seguir para instalar en el robot el firmware que permite el uso del lenguaje Python se adjunta en la parte de anexo.

Una vez realizado lo descrito en el apéndice podemos proceder a realizar los siguientes pasos, primero debemos comunicar dicho robot a través del USB o bien de forma inalámbrica con el XBee el cual se describe sus características en la sección anexo.

Primero es necesario conectar el cable ó el módulo de comunicaciones a la computadora, utilizando para esto alguno de los puertos USB disponibles. Al hacerlo, se creará un dispositivo con un nombre similar a

`/dev/ttyUSB0`

en el sistema, el número (en este caso 0) puede variar. Podemos comprobar esto desde una terminal de la siguiente manera:

```
usuario@host:~$ ls /dev/ttyUSB*/dev/ttyUSB0* /dev/ttyACM0
```

Estos son las 3 formas posibles que podría aparecer como opción siendo este último (dev/ttyACM0) el que corresponde al sistema linux debían 8 que utilizaremos. Luego es necesario encender el robot. Una vez presionado el botón I/O para encender el robot, es necesario esperar a que el LED de estado naranja de la esquina inferior derecha deje de parpadear. Luego, hay que presionar el botón “Run”, cuando este LED se apague podremos empezar a usar el robot. Podemos ver la ubicación de estos botones en la Figura 2 y el LED de estado que se presentan a continuación.

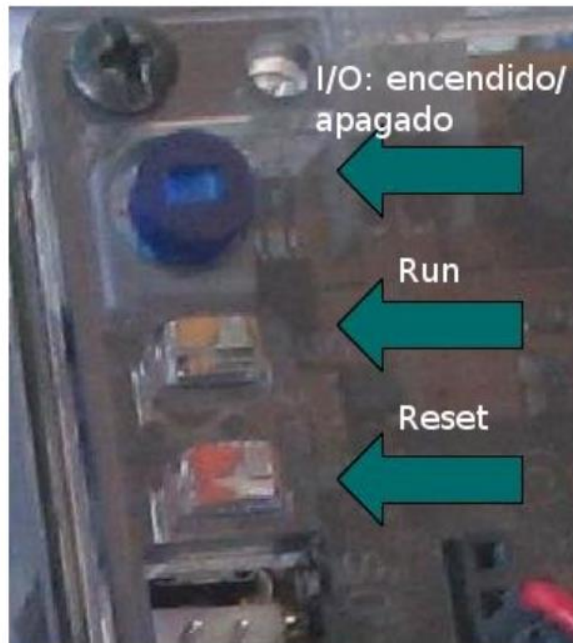


Figura 2- Botones N6

Pero antes de trabajar con el robot debemos importar las funciones del mismo, esto se debe a que las funciones propias del robot no vienen integradas con el lenguaje Python y, para poder usarlas, debemos incorporarlas al entorno. Esto se hace utilizando la sentencia import del lenguaje Python:

```
>>> from duinobot import *
```

Una vez realizado lo anterior, ya podemos comenzar a utilizar el robot. Lo primero que debemos hacer, es identificarlo de alguna manera. Para eso, debemos ubicar cuáles robots se encuentran prendidos y conectados a nuestra computadora. Esto lo hacemos de la siguiente manera:

```
[1] >>> b = Board("/dev/ttyACM0")
```

```
[2] >>> b.report()
```

```
[3] [1, 2, 3]
```

Para realizar los pasos descriptos se debe de conocer la ruta donde este conectado el robot.

Una vez que el robot se halla comunicado podemos proceder a desarrollar scripts en Python, a continuación se deja uno expresado.

```
import time
import random

1. from duirobot import *
2. board = Board("/dev/ttyUSB0")
3. board.report()
4. robot = Robot(board, 5)
5. try:
6.     while True:
7.         robot.forward(10)
8.         print "Distancia:", robot.ping()
9.         robot.bEEP()
10.        if robot.getObstacle():
11.            print "choque!"
12.            angulo = random.randint(0,100)
13.            print "girando", angulo, "grados!"
14.            robot.turnleft(angulo, 0.5)
15.            time.sleep(1)
16. finally:
17.     robot.stop()
18.     board.exit()
```

Figura 3- Script robot

El código presentado en la Figura 3 lo que hace es, hacer avanzar el robot haciendo que el mismo tome decisiones “propias” de girar a la hora de toparse con algún obstáculo, como se menciona con anterioridad, se necesita conocer la ruta por donde esté conectado el robot para poder transmitir los datos del mismo.

3. Conclusiones

- Se concluye de forma parcial que el lenguaje propuesto para la modificación del firmware es el adecuado, no se obtuvo problemas mayores, también podemos reflexionar que por el momento el proyecto se encuentra en proceso de desarrollo, a futuro se espera expandir estos conceptos al campo de otros sistemas operativos como ser Windows, siendo este uno de los mas usados.

Referencias

1. Octavio J Da Silva Gilling, Monica J Paves Palacios, Julian Da Silva Gilling. Curso de Robótica. Copyright (c) 2009 Multiplo® <http://multiplo.org> IMPRESO EN ARGENTINA POR ROBOTGROUP®
<http://robotgroup.com.ar> <http://roboticaeducativa.net>
2. DATASHEET ATmega16U4/ATmega32U4 2015 Atmel Corporation.
http://ww1.microchip.com/downloads/en/devicedoc/atmel-7766-8-bit-avr-atmega16u4-32u4_datasheet.pdf
3. 8-bit Microcontroller with 128K Bytes In-System Programmable Flash ATmega1284P .2009 Atmel Corporation. <https://ww1.microchip.com/downloads/en/DeviceDoc/doc8059.pdf>
4. Historia del Software: el lenguaje Python. Dic 9, 2011 - 21:28 (CET)_Autor: [jjvelasco](#).
<https://hipertextual.com/archivo/2011/12/lenguaje-python/>
5. Pagina ofical de python. Paython Software Fundation <https://www.python.org/psf/>
6. Manual de programación con robots para la escuela. Claudia Banchoff, Joaquín Bogado, Damián Mel, Sofía Martin, Fernando López. Agosto 2012 - Versión 0.1.5
7. Aprenda a Pensar Como un Programador con Python, Allen Downey, Jeffrey Elkner, Chris Meyers. Green Tea Press, Wellesley, Massachusetts. Versión PDF.
Copyright @ 2002 Allen Downey, Jeffrey Elkner, y Chris Meyers.
8. Programacion procedural universidad de Oviedo dpto de Informática.
<http://www.atc.uniovi.es/telematica/2ac/Transparencias/T02-Programacion-Procedural.pdf>
9. ¿Qué es xbee? Página oficial Xbee. <https://xbee.cl/que-es-xbee/>

10. Einar Felipe Lanfranco - Joaquin Bogado David Vilaseca vilaseca Julian da Silva
Gilligjulian Modificaciones realizadas al Robot Multiplo N6 para permitir programación interactiva.

11. Ortega Huembes, Carlos Alberto (2008).
«Zigbee: El nuevo estándar global para la domótica e inmótica».
Consultado el 10 de febrero de 2015

12. Instalación del firmware para usar los N6 con Python Sofía Martin, Fernando López –
Linti – UNLP 27 de agosto de 2014

13. Pagina web Wikipedia. https://en.wikipedia.org/wiki/IEEE_802.15.4

14. Pagina web Wikipedia. <https://es.wikipedia.org/wiki/Zigbee>

15. Ingeniería MCI Ltda web oficial de XBEE <https://xbee.cl/que-es-xbee/>

16. Ingeniería MCI Ltda web oficial de XBEE. © 2020 Digi International Inc. All rights reserved.
Digi XBee® 3 Zigbee® RF Module updated on 08 Oct 2019 08:35 AM
https://www.digi.com/resources/documentation/Digidocs/9000194213/concepts/c_xbee_antennas.htm?TocPath=Additional%20resources%7CXBee%20buying%20guide%7C_____2

17. [Vázquez Fernández-Pacheco, A. S. Ramos de la Flor, F. y Fernández Rodríguez, R. \(2015\). Robótica educativa. RA-MA Editorial. Recuperado de https://elibro.net/es/ereader/elibrounam/106572?page=77.](https://elibro.net/es/ereader/elibrounam/106572?page=77)

Anexo

Describiremos como instalar el firmware (12.) usando Debian GNU/Linux. Primero es necesario instalar algunos paquetes que permitirán descargar y hacer funcionar nuestra versión modificada de DuinoPack:

Ingresar e modo usuario con el comando “su”

```
# apt-get install git
```

```
$ git clone https://github.com/Robots-Linti/v1.2.git duinopack
```

Con esta orden se descarga e instalan los paquetes correspondientes a duinopack modificados.

También es necesario descargar el firmware N6Firmata también desde GitHub:

```
$ git clone https://github.com/Robots-Linti/n6firmata.git
```

Aqui se descarga e instala la librería N6firmata que se instalará en el robot para que realice la interpretación de los comandos Python.

Finalmente ejecutar DuinoPack (IDE Arduino modificado):

```
$ cd duinopack/linux/DuinoPack.v1.2.Linux/
```

Este paso puede realizarse en el entorno gráfico si nos dirigimos a la capeta donde está instalado el programa Figura 1, hacer un clic derecho seleccionar abrir un terminal y proseguir con el siguiente paso.

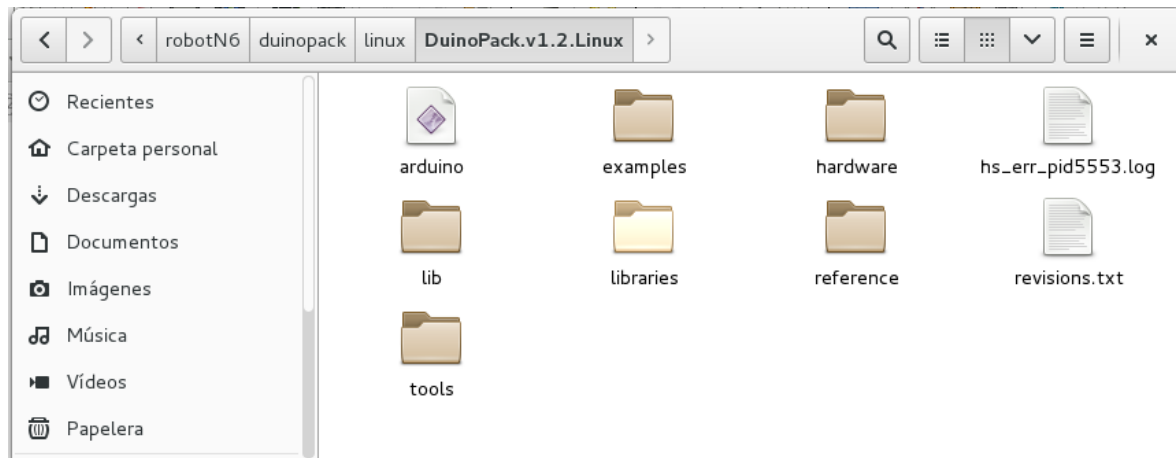


Figura 1 carpeta donde fue instalado el arduino ide.

Escribimos en un terminal de linux el siguiente código para que se abra el arduino ide:

```
$ arduino
```

Luego se selecciona el puerto donde se conecta la placa. Figura Figura 2. Si no está activo es porque no poseen los permisos de lectura/escritura sobre los mismos y hay que concederlos, para esto utilizamos el comando chmod de la siguiente manera:

```
$chmod 777 /dev/ttyACM0
```

Con este comando quedan concebidos los permisos necesarios para la lectura/escritura en el puertoUSB.

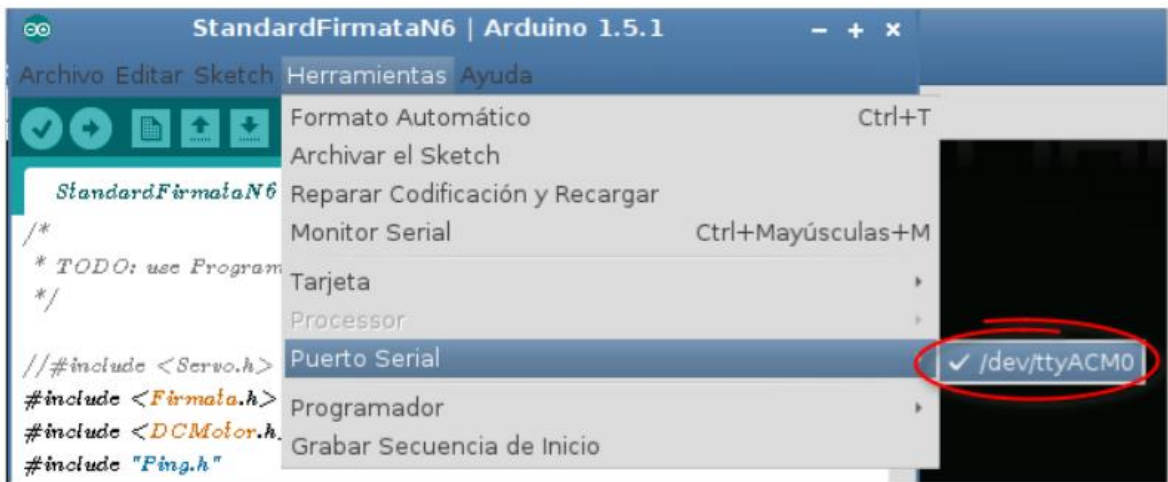


Figura 2 Selección del puerto al que está conectada la placa

1. Comprobación de las comunicaciones

Para corroborar el correcto funcionamiento de la comunicación entre el robot y el arduino puede realizarse el siguiente paso, si no se desea realizar esta comprobación puede obviarse sin ningún perjuicio. Dependiendo la versión instalada puede procederse de varias maneras, realizaremos la explicación de dos de ellas.

Primero: realizamos la carga del ejemplo “hello world” vamos al menú Archivo, ejemplos, 00.multiplo, Hello World.

Puede que la versión descargada no contenga este ejemplo, se lo puede bajar de la red y abrir como un archivo común tipo .ino.

Otra opción sería escribirlo ya que no es un programa extenso:

```
void setup()  
{  
  delay(1000);  
  Serial.begin(9600);  
  Serial.println("Hello World!");  
}  
  
void loop()  
{}
```

Se realiza la carga del programa como se muestra en la figura 3 y se procede a abrir un monitor serial, este se encuentra en el menú herramientas y e muestra en la figura 4. Si está todo funcionando correctamente se verá lo que responde el robot, saludándonos.



Figura 3 carga del ejemplo

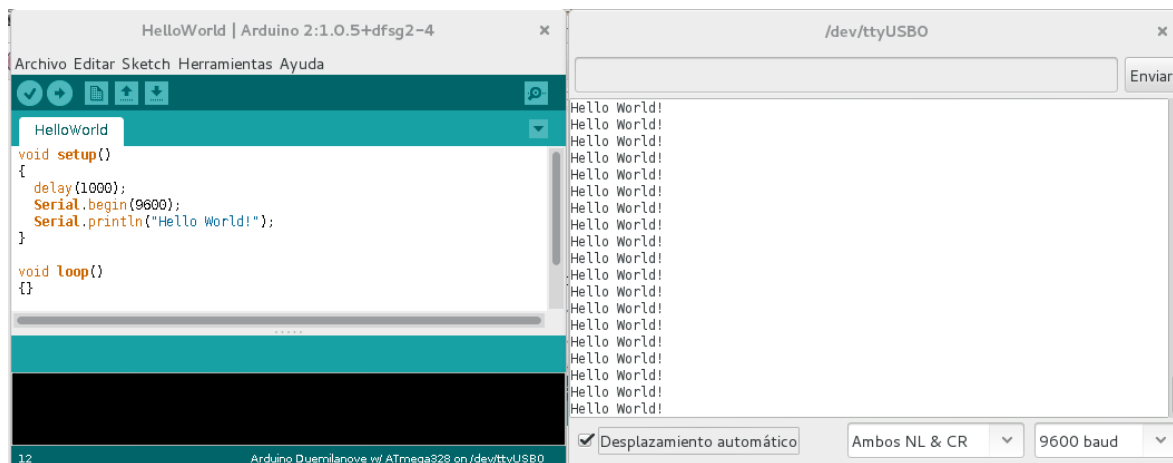


Figura 4 Monitor serial en funcionamiento

1. Carga del interprete para trabajar con python

Ahora que esta todo listo y sabemos que la memoria del robot se carga correctamente se procede a abrir el intérprete que será cargado a la memoria del N6. Esta es la dirección del archivo **n6firmata/StandardFirmataN6/StandardFirmataN6.ino** desde el IDE DuinoPack, seleccionar el tipo de tarjeta “Duinobot.v1.x” en el menú Herramientas->Tarjeta como se ve en la Figura Figura 5. En el caso de la versión 2.4 de duinobot, el tipo de tarjeta es “Arduino Diecimila or Duemilanove w/ Atmega328” en el mismo menú tarjeta

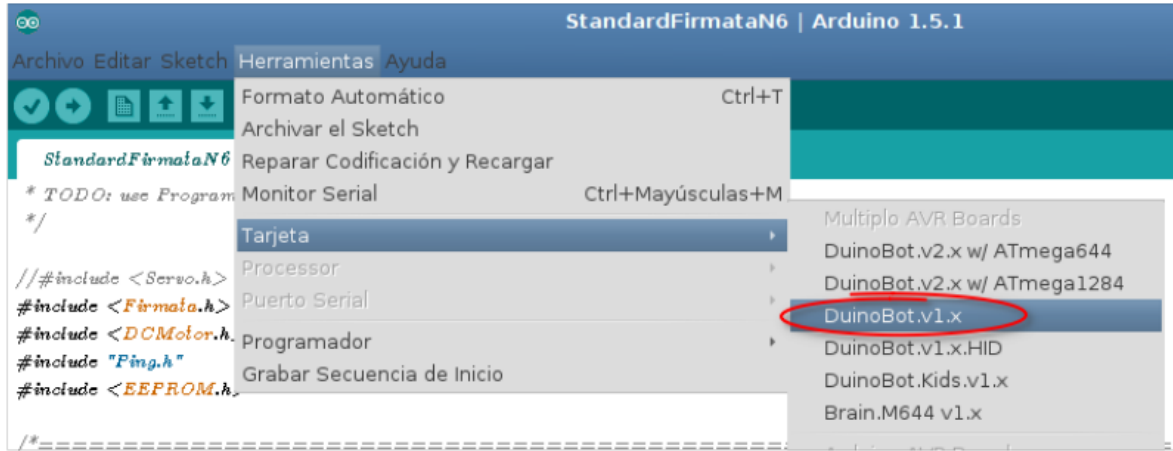


Figura 5 Seleccionar Tarjeta

Luego hacer upload de StandardFirmataN6.ino Figura Figura 6

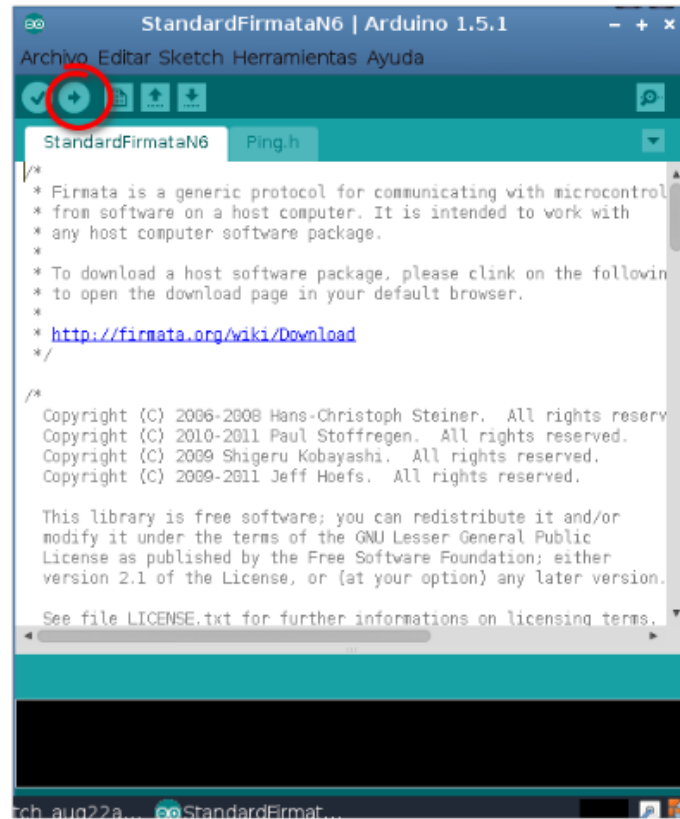


Figura 6 Upload

Con esas configuraciones el robot queda en condiciones para recibir y enviar información al programa realizado en python