

Arduino Programming in Python

Sergio Paniego Blanco



Who am I?



Artificial Intelligence MSc - UPM

**Informatic Engineering and Software
Engineering - URJC**

Student Developer GSoC - JdeRobot



sergiopaniego



sergiopaniego



sergiopaniego



Summary

Google Summer of Code

Problematic

State of the art

PyOnArduino

Demo



Google Summer of Code

Global program

Bring students into open source development

Remote work + meetings

JdeRobot – 6 students

Timeline

- February - Organizations announced
- March - Student applications
- April – Student Projects announced
- April & May – Community Boarding
- May to August – Coding
- June, July & August - Evaluations

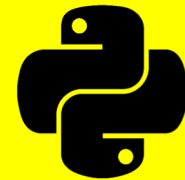


Problematic

Arduino programming language complexity

Need of easy to use tools for newbies

Code Arduino boards using Python





State of the art

Current solution for the problematic

Continuous communication between the robot and the PC.

Possible approaches

- Pyxie
- Cython
- LLVM





PyOnArduino

People:

José María Cañas

Gorka Guardiola

Luis Roberto Morales

Sergio Paniego Blanco

JdeRobot Robot Programming tool

Translate Python-like code to Arduino code

Translate, compile and upload the code directly


```
def loop():  
    if halduino.getLineFollowValue() == 0:  
        halduino.setSpeedEngines(100, 100)  
    elif halduino.getLineFollowValue() == 1:  
        halduino.setSpeedEngines(0, 100)  
    elif halduino.getLineFollowValue() == 2:  
        halduino.setSpeedEngines(100, 0)  
    elif halduino.getLineFollowValue() == 3:  
        halduino.setSpeedEngines(-100, -100)
```





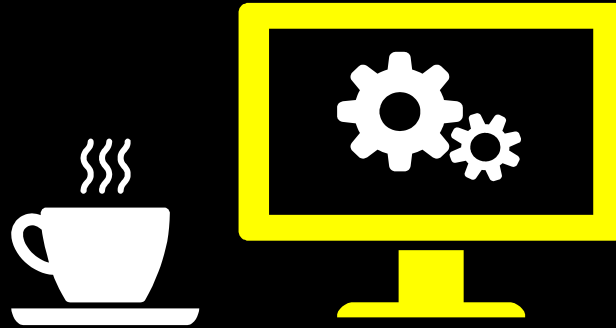
Supported robots

MBOT



COMPLUBOT





PyOnArduino's structure





Translator

Abstract Syntax Tree

Each node – Statement in the code

Information for analyzing the code

Inessential punctuation and delimiters

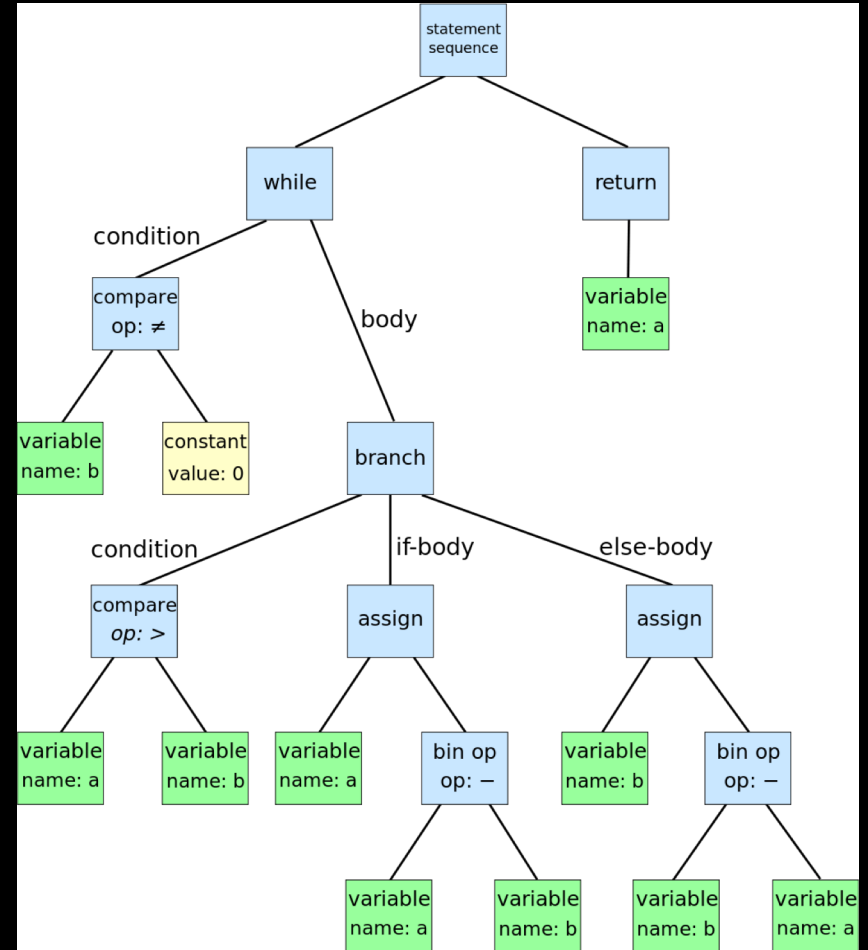
Python library

NodeTransformer and NodeVisitor

Abstract Syntax Tree

Greatest Common Divisor

```
while b ≠ 0
  if a > b
    a := a - b
  else
    b := b - a
return a
```





Translator

Abstract Syntax Tree

Each node – Statement in the code

Information for analyzing the code

Python library

NodeTransformer and NodeVisitor

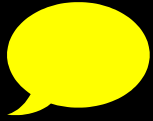
NodeVisitor & NodeTransformer

```
import ast

class MyVisitor(ast.NodeVisitor):
    def visit_Str(self, node):
        print('String Node: "' + node.s + '"')

class MyTransformer(ast.NodeTransformer):
    def visit_Str(self, node):
        return ast.Str('str: ' + node.s)

parsed = ast.parse("print('Hello World')")
MyTransformer().visit(parsed)
MyVisitor().visit(parsed)
```

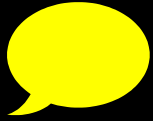
HALduino

Hardware Abstraction Layer

halduino.py

Specific halduino for each robot

Get rid of Arduino's complexity



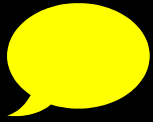
HALduino example

Python

```
setSpeedEngines(leftSpeed: int, rightSpeed: int)
```

Arduino

```
MeDCMotor leftMotor(9);  
MeDCMotor rightMotor(10);  
void setSpeedEngines(int speedLeft, int speedRight) {  
    leftMotor.run(speedLeft);  
    rightMotor.run(speedRight);  
}
```



Sensors and actuators supported

Complubot

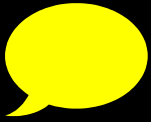
| Sensor/Actuator | Supported functions |
|--------------------|------------------------------------|
| DC Engines | setSpeedEngines (left,right) |
| Ultrasonic sensors | getUS() |
| Infrared sensors | getIR[1,2,3,4,5]() |
| Beep emitter | playBeep(type) |
| Sound emitter | playMelody(melody) |
| Screen write | setScreenText(text), cleanScreen() |



Sensors and actuators supported

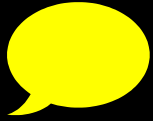
mBot

| Sensor/Actuator | Supported functions |
|--------------------|----------------------------------------|
| DC Engines | setSpeedEngines (speed) |
| Ultrasonic sensors | getUS() |
| LEDs | setLeds(ledNumber, red, green, blue) |
| Infrared sensors | getMessage(), sendMessage(message) |
| Light sensor | getLightSensor() |
| Button | isButtonPressed(), isButtonReleased() |
| Buzzer | playBuzzer(tone, length) |
| External screen | drawString(name), showClock(hour, min) |



Python features supported

| Feature | Limitations/Comments |
|----------------------|-------------------------------|
| Variable declaration | SUPPORTED |
| Function declaration | With/without return statement |
| Operators | + - / * ^ % |
| Comparators | < <= >= == != |
| Logic operators | And or is not |
| pass | SUPPORTED |
| loops | While, for(limited) |
| Sleep() | SUPPORTED |
| if | If/elif/else |
| Boolean operations | and or |
| print | SUPPORTED |

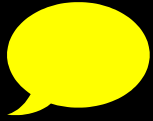


Stop and go code example Mbot

```
import HALduino.halduino as halduino

def set_engine(direction: int):
    if direction == 0:
        halduino.setSpeedEngines(0, 0)
        print('STOP!')
    elif direction == 1:
        halduino.setSpeedEngines(100, 100)
        print('Forward')

def loop():
    if halduino.getUS() < 10:
        set_engine(0)
    else:
        set_engine(1)
```



Stop and go code example Complubot

```
import HALduino.halduino as halduino

def set_engine(direction: int):
    if direction == 0:
        halduino.setSpeedEngines(0, 0)
        print('STOP!')
    elif direction == 1:
        halduino.setSpeedEngines(100, 100)
        print('Forward')

def loop():
    if halduino.getUS() < 30:
        set_engine(0)
    else:
        set_engine(1)
```



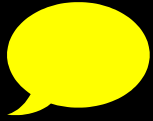
Managing robot's architecture

Versatility

Configure robot's ports

```
# Example of an mBot setup
```

```
leftMotor = 9  
rightMotor = 10  
ultrasonicSensor = 3  
rgbled = 7  
lightSensor = 6  
ledMtx = 3  
lineFollower = 2
```

Executing PyOnArduino

Python 3.x

Arduino IDE

Arduino Makefile

```
python3 translator/Translator.py [input-file] [robot]  
python3 translator/Translator.py [input-file] [robot] [architecture-file]
```



Make & Arduino Makefile

Build automation

Directives used by make

make upload

```
# mBot Makefile

ARDUINO_LIBS= Makeblock-Libraries-master Wire SPI
MONITOR_PORT= /dev/cu.wchusbserial1420
BOARD_TAG = uno
ARDUINO_DIR  = /Applications/Arduino.app/Contents/Java
include /usr/local/opt/arduino-mk/Arduino.mk
```



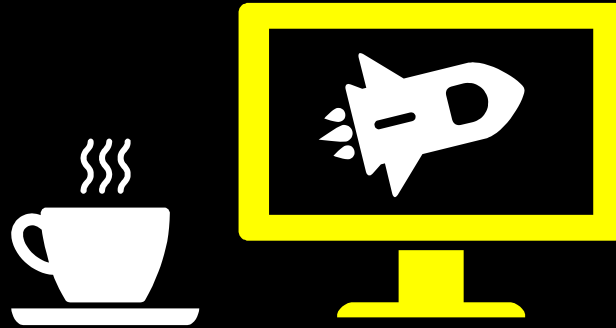
Problems during development

Dynamic typing in Python vs Arduino

Architectural Stop

Variables' type in function declaration

Lost parentheses



DEMO TIME!

Lets see how PyOnArduino tool works!

Code

Issues 1

Pull requests 0

Projects 0

Wiki

Insights

No description, website, or topics provided.

169 commits

1 branch

0 releases

1 environment

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

| | |
|---------------------------------------------------------|------------------------------------------------------------------------------|
| sergiopaniego Melody test for Complubot restored | Latest commit d84018d a day ago |
| .idea | Architectural error for Complubot motor 3 months ago |
| HALduino | Architectural error for Complubot motor 3 months ago |
| docs | Hola gif added 3 months ago |
| examples | Melody test for Complubot restored a day ago |
| makefiles | mBot tests added and comments in files 3 months ago |
| tests | Exception management improved for non-implemented functionality 3 months ago |
| translator | Exception management improved for non-implemented functionality 3 months ago |
| .gitignore | Small example generated using ast 5 months ago |
| README.md | README.md updated 14 days ago |

README.md

PyOnArduino

Thanks!

Any questions?

You can find me at
@sergiopaniego