

**UNIVERSIDADE FEDERAL DO PARANÁ**

**AGRONOMIA**

**ALUNAS: JÉSSICA SUELEN DE SOUZA GRANJA**

**THAIS BUWAI LUCIF**

**UTILIZAÇÃO DE PROGRAMAÇÃO EM PYTHON PARA CONTROLE DE  
SENSORES DE UMIDADE DO SOLO**

**CURITIBA**

**2018**

JÉSSICA SUELEN DE SOUZA GRANJA – GRR20185539

THAIS BUWAI LUCIF – GRR20180877

**UTILIZAÇÃO DE PROGRAMAÇÃO EM PYTHON PARA CONTROLE DE  
SENSORES DE UMIDADE DO SOLO**

Relatório apresentando à  
disciplina de Fundamentos da  
programação de computadores do  
curso de graduação de Agronomia da  
Universidade Federal do Paraná.

Prof. Jackson Antônio do Prado Lima

CURITIBA

2018

2

## SUMÁRIO

INTRODUÇÃO.....	Pág. 4
OBJETIVO.....	Pág. 5
DESENVOLVIMENTO.....	Pág. 6
RESULTADOS E DISCUSSÕES.....	Pág.11
OBSERVAÇÃO.....	Pág. 13
REFERÊNCIAS .....	Pág. 14



## **INTRODUÇÃO**

Este relatório contém informações sobre o trabalho realizado em Python para a utilização de sensores de umidade do solo e ar para controle de irrigação com o auxílio do Raspberry Pi3.

## **OBJETIVOS**

O principal objetivo é criar um conjunto de programas que auxiliem o Raspberry Pi3 a identificar por meio de sensores de umidade o momento em que o solo está seco para que seja feita a irrigação.

## **DESENVOLVIMENTO**

Para o desenvolvimento do trabalho faz-se necessário o uso de alguns materiais para a montagem do sistema que será listado abaixo.

### **Raspberry Pi3**

Um microcomputador desenvolvido no Reino Unido pela fundação Raspberry Pi com principal objetivo de ensinar computação nas escolas. Permite a exploração da computação tendo como principal linguagem utilizada Python através do programa Python 3 IDLE.



Foto: Jéssica Suelen

### **Sensor de umidade do solo: módulo e sonda.**

Utilizaremos o potenciômetro para regular um valor e receber um sinal (variando de 0 a 1) na saída analógica ou valores detalhados de umidade do solo a partir de uma saída digital.

Neste caso foi utilizada a saída analógica devido a falta de um conversor.

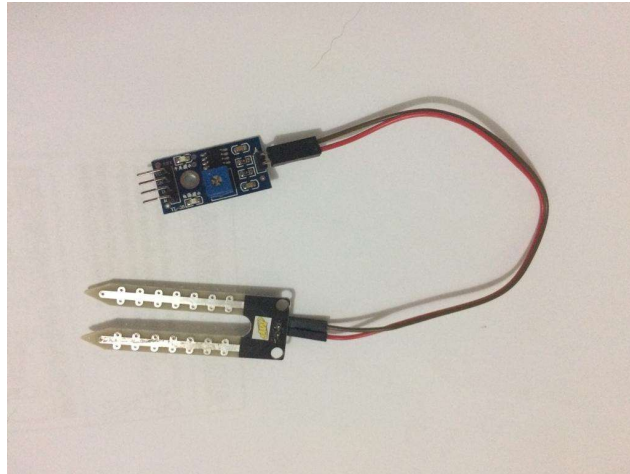


Foto 1.1: Jéssica Suelen - sensor de umidade do solo.

### Protoboard e conexões.

Ferramenta importante para montagem dos circuitos pois não necessita solda dos componentes facilitando a execução de trabalhos.

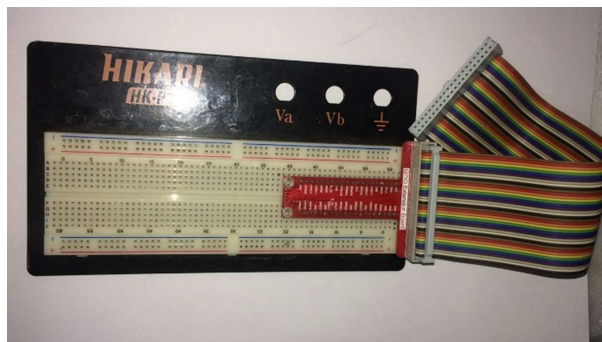


Foto 2 : Jéssica Suelen – Protoboard e conexões

### Válvula solenoide

Válvula eletromecânica acionada a fim de controlar o fluxo de líquidos ou gases.



Foto 3: Jéssica Suelen - Válvula



## Relé

Para transformação da corrente elétrica (5v);



Foto 4: Jéssica Suelen – relé

**E por último, uma fonte qualquer 12v para ligar a válvula.**

Para a utilização do programa é necessário que seja instalada todas as bibliotecas utilizadas no Raspberry. Primeiramente utilizaremos a biblioteca RPi.GPIO no Python, dentro desta biblioteca há uma função denominada `event_callback` que retorna uma função definida pelo usuário quando houver diferenças e algum pino no qual está ligado o sensor.

```
lrriga_Novo2.py - /home/pi/lrriga_Novo2.py (3.5.3)
File Edit Format Run Options Window Help
# Definindo Bibliotecas GPIO
import RPi.GPIO as GPIO
from time import sleep
#
GPIO.setmode(GPIO.BOARD) # Define o número de ordem do pino
GPIO.setwarnings(False) # Apaga Avisos
#
# Definindo Portas de Entrada e Saída
#
GPIO.setup(35, GPIO.IN)
GPIO.setup(37, GPIO.OUT)
#
# Rotina para mudar de comando quando há
# variação de estado no pino de entrada
#
def mudou(channel):
    if GPIO.input(35):
        print ("SOLO SECO - IRRIGAÇÃO ACIONADA ")
        GPIO.output(37,1)
    else:
        print ("SOLO UMIDO - IRRIGAÇÃO DESLIGADA")
        GPIO.output(37,0)
#
# Detecção de Variação no pino de Entrada
#
GPIO.add_event_detect(35, GPIO.BOTH, callback=mudou)
```

Fig1.: Configuração do sensor de umidade do solo.

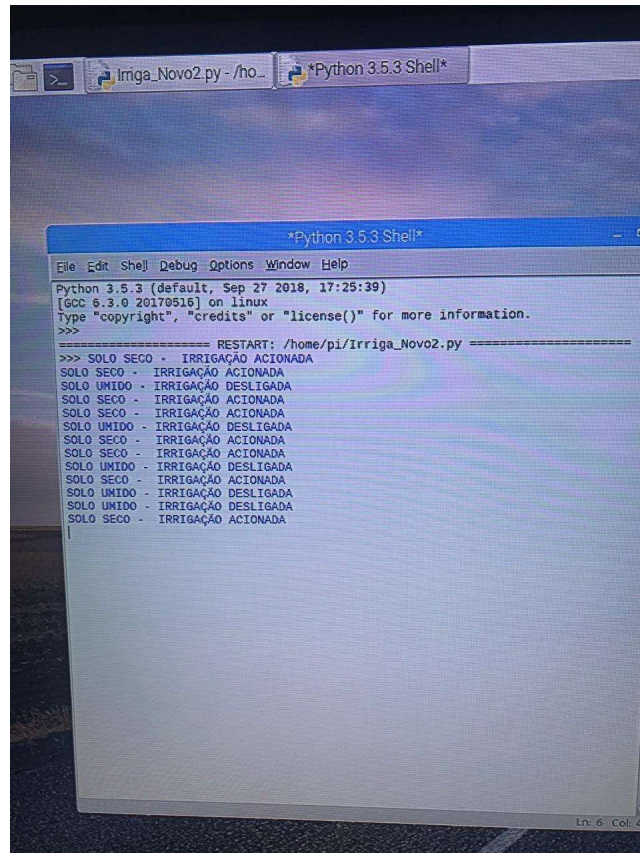
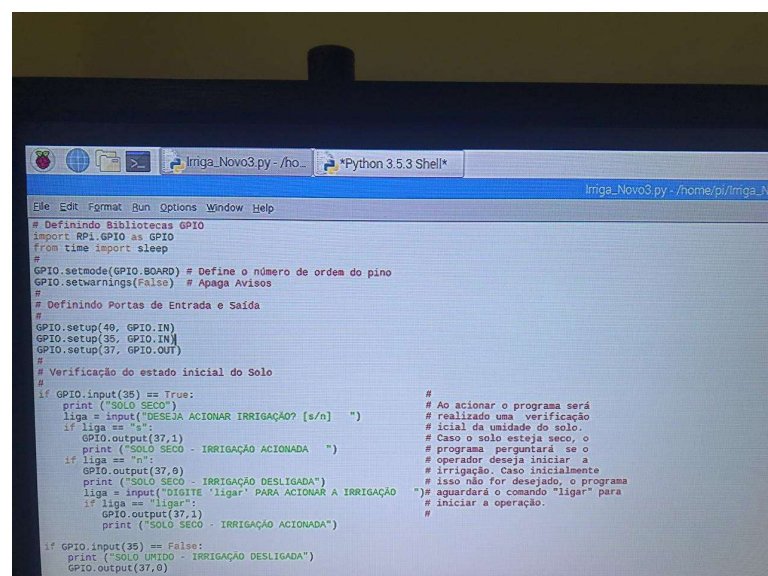


Fig1.1: Teste do programa

A partir deste código, o programa irá indicar quando precisa ou não irrigar.

## Para o acionamento do relé e válvula



```
# GPIO.input(35) == False:
print ("SOLO UMIDO - IRRIGAÇÃO DESLIGADA")
GPIO.output(37,0)

#
# Rotina para mudar de comando quando há
# variação de estado no pino de entrada
#
def mudou(channel):
    if GPIO.input(35) == True:
        print ("SOLO SECO - IRRIGAÇÃO ACIONADA ")
        GPIO.output(37,1)

    if GPIO.input(35) == False:
        print ("SOLO UMIDO - IRRIGAÇÃO DESLIGADA ")
        GPIO.output(37,0)

#
# Rotina para Parada de Emergência
#
def parou_tudo(channel):
    if GPIO.input(35) == True:
        print ("PARADA DE EMERGENCIA")
        sleep(5)
        GPIO.output(37,0)
        print ("IRRIGAÇÃO CANCELADA")

#
# Detecção de Variação no pino de Entrada
#
GPIO.add_event_detect(35, GPIO.BOTH, callback=mudou)

#
# Parada de Emergência
GPIO.add_event_detect(40, GPIO.FALLING, callback=parou_tudo)
#
```

The image shows a computer monitor with a Python 3.5.3 Shell window open. The window's title bar indicates the current file is 'Irriga\_Novo3.py' located in the directory '/ho...'. The menu bar contains options: File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following text:

```
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170518] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/pi/Irriga_Novo3.py =====
SOLO SECO
DESEJA ACIONAR IRRIGAÇÃO? [s/n] s
SOLO SECO - IRRIGAÇÃO ACIONADA
>>> SOLO SECO - IRRIGAÇÃO ACIONADA
SOLO UMIDO - IRRIGAÇÃO DESLIGADA
SOLO UMIDO - IRRIGAÇÃO DESLIGADA
SOLO SECO - IRRIGAÇÃO ACIONADA
SOLO SECO - IRRIGAÇÃO ACIONADA
SOLO UMIDO - IRRIGAÇÃO DESLIGADA
SOLO UMIDO - IRRIGAÇÃO DESLIGADA
SOLO SECO - IRRIGAÇÃO ACIONADA
SOLO SECO - IRRIGAÇÃO ACIONADA
SOLO UMIDO - IRRIGAÇÃO DESLIGADA
SOLO UMIDO - IRRIGAÇÃO DESLIGADA
SOLO SECO - IRRIGAÇÃO ACIONADA
```

To the right of the main window, a portion of another application window titled 'gramma' is visible, containing fragments of code such as 'erifi', 'do s', 'ja se', 'tará', 'inicia', 'iniciã', 'ejado', 'ndo "l', and 'ão.'

11



O programa permite que ao detectar a ausência de umidade no solo, seja enviado um aviso seguido de uma pergunta (deseja acionar a irrigação?). Desta forma, o usuário pode interferir no processo, caso queira retirar o sensor do solo, por exemplo.

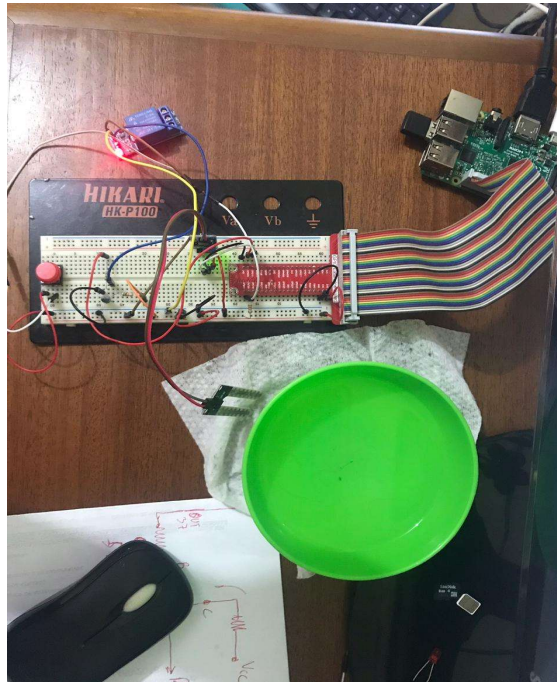


Fig. 3 – Sistema e conexões montadas – parte elétrica.

## RESULTADOS E DISCUSSÕES

Através do sistema Raspberry é possível criar um sistema com programação em Python para irrigação automatizada. Entretanto, é um sistema considerado caro, pois exige a presença de um computador ligado constantemente.

Nesses casos, o Arduino Uno é mais adequado e mais barato, pois só precisa de um computador para programa – lo. O mesmo não foi utilizado neste trabalho pois a linguagem computacional utilizada é a “C”.

A disciplina de Fundamentos da programação de computadores permitiu que conseguíssemos entender o conceito de funções, de como importar ferramentas de bibliotecas, do comando while e condicionais (if, else) para que fosse executado de forma correta, sendo as principais funções utilizadas neste projeto, tendo em vista que os dados são obtidos através dos próprios sensores.



**Observação:** A ligação da válvula a mangueira de irrigação não foi realizada devido a pouco conhecimento de hidráulica, porém é possível a ligação pois toda a parte eletrônica apresentou perfeito funcionamento durante os testes.

Segue abaixo o código em python utilizado:

```
gh x
1  # Definindo Bibliotecas GPIO
2  import RPi.GPIO as GPIO
3  from time import sleep
4  #
5  GPIO.setmode(GPIO.BOARD) # Define o número de ordem do pino
6  GPIO.setwarnings(False) # Apaga Avisos
7  #
8  # Definindo Portas de Entrada e Saída
9  #
10 GPIO.setup(40, GPIO.IN)
11 GPIO.setup(35, GPIO.IN)
12 GPIO.setup(37, GPIO.OUT)
13 #
14 # Verificação do estado inicial do Solo
15 #
16 if GPIO.input(35) == True:
17     print ("SOLO SECO")
18     liga = input("DESEJA ACIONAR IRRIGAÇÃO? [s/n] ")
19     if liga == "s":
20         GPIO.output(37,1)
21         print ("SOLO SECO - IRRIGAÇÃO ACIONADA ")
22     if liga == "n":
23         GPIO.output(37,0)
24         print ("SOLO SECO - IRRIGAÇÃO DESLIGADA")
25     liga = input("DIGITE 'ligar' PARA ACIONAR A IRRIGAÇÃO ")
26     if liga == "ligar":
27         GPIO.output(37,1)
28         print ("SOLO SECO - IRRIGAÇÃO ACIONADA")
29 #
30 if GPIO.input(35) == False:
31     print ("SOLO UMIDO - IRRIGAÇÃO DESLIGADA")
32     GPIO.output(37,0)
33 #
34 # Rotina para mudar de comando quando há
35 # variação de estado no pino de entrada
36 #
37 def mudou(channel):
38     if GPIO.input(35) == True:
39         print ("SOLO SECO - IRRIGAÇÃO ACIONADA ")
40         GPIO.output(37,1)
41     if GPIO.input(35) == False:
42         print ("SOLO UMIDO - IRRIGAÇÃO DESLIGADA ")
43         GPIO.output(37,0)
44 #
45 # Rotina para Parada de Emergência
46 #
47 def parou_tudo(channel):
48     if GPIO.input(35) == True:
49         print ("PARADA DE EMERGÊNCIA")
50         sleep (5)
51         GPIO.output(37,0)
52         print ("IRRIGAÇÃO CANCELADA")
53 #
54 # Rotina para Parada de Emergência
55 #
56 def parou_tudo(channel):
57     if GPIO.input(35) == True:
58         print ("PARADA DE EMERGÊNCIA")
59         sleep (5)
60         GPIO.output(37,0)
61         print ("IRRIGAÇÃO CANCELADA")
62 #
63 # Detecção de Variação no pino de Entrada
64 #
65 GPIO.add_event_detect(35, GPIO.BOTH, callback=mudou)
66 #
67 # Parada de Emergência
68 GPIO.add_event_detect(40, GPIO.FALLING, callback=parou_tudo)
69 #
```

## Referências Bibliográficas

**Fazedores:** <http://blog.fazedores.com/raspberry-pi-b-introducao-porta-gpio/> - Acesso em 24 de novembro de 2018.

**Python Brasil:** <https://wiki.python.org.br/AprendaProgramar> - Acesso em 24 de novembro de 2018.

**Excript:** <http://excript.com/python/while-else-python.html> - acesso em 23 de novembro de 2018.

**Rede agronomia:** <http://agronomos.ning.com/profiles/blogs/python-essa-linguagem-maravilhosa> acesso em 24 de novembro de 2018.

Raspi.tv: <https://raspi.tv/2014/rpi-gpio-update-and-detecting-both-rising-and-falling-edges> - acesso em 24 de novembro de 2018.