

Automatização de Processos Industriais Utilizando Raspberry Pi

Discente: Ethierre Rockenbach Feldkircher

RGa:201711902038

Discente: Rafael Fernando Bezerra

RGa: 201621902004

Discente: Rayan Victor Souza Aguiar

RGa:201711902032

Disciplina:

MICROCONTROLADORES E SISTEMAS DIGITAIS

Prof. Dr. Jesús Franco Bueno

Cuiabá, agosto de 2019.

Automatização de Processos industriais utilizando Raspberry Pi

MICROCONTROLADORES E SISTEMAS DIGITAIS

Relatório Executivo do projeto com Microcontroladores e apresentado na disciplina de Microcontroladores e Sistemas Digitais como parte da avaliação do aprendizado.

ÍNDICE DE FIGURAS

Figura 1 - Componentes essenciais de um Projeto com Microcontrolador	6
Figura 2 – Especificação dos pinos GPIO disponíveis na Raspberry Pi	8
Figura 3 – Componente físico do atuador servo motor 5V	9
Figura 4 – Componente físico do atuador motor DC 12V..Error! Bookmark not defined.0	
Figura 5 – Componente driver motor ponte h l298n	110
Figura 6 – Montagem do esquema para programação e teste do atuador.....	141
Figura 7 – Componente físico do atuador com motor	144
Figura 8 – Montagem do esquema para programação e teste do atuador.....	144
Figura 9 – Interface de Programação para Microcontrolador - IDLE Python	14

SUMÁRIO

1	APRESENTAÇÃO DESTE TRABALHO – TEXTO DO PROFESSOR DA DISCIPLINA.....	5
2	RESUMO EXECUTIVO DO PROJETO	6
3	PARTE 1 – CONCEITOS ESSENCIAIS PARA PROJETO COM MCU	6
3.1	COMPONENTES DA ARQUITETURA DE UM PROJETO COM MICROCONTROLADOR	6
3.2	DESCRIÇÃO CONCEITUAL DOS COMPONENTES DA ARQUITETURA DO PROJETO.....	7
3.2.1	<i>Interface de usuário.....</i>	<i>7</i>
3.2.2	<i>Microcontrolador escolhido.....</i>	<i>7</i>
3.2.2.1	<i>Entradas/saídas do Microcontrolador</i>	<i>7</i>
3.2.3	<i>Sensor.....</i>	<i>8</i>
3.2.4	<i>Atuador.....</i>	<i>8</i>
3.2.5	<i>Alimentação e Refrigeração.....</i>	<i>9</i>
4	PARTE 2 – SIMULAÇÃO E/OU MONTAGEM DO PROJETO COM MCU	9
4.1	DESCRIÇÃO DO PROJETO ESCOLHIDO PARA EXECUÇÃO COM MCU	9
4.2	O MICROCONTROLADOR DO PROJETO	9
4.2.1	<i>Implementação da Alimentação e Refrigeração.....</i>	<i>9</i>
4.3	PROTOTIPAÇÃO PARA TESTE, CALIBRAÇÃO E AFERIÇÃO DO SENSOR.....	10
4.3.1	<i>Esquemático para conexão do sensor FSR</i>	<i>11</i>
4.3.1.1	<i>Código-fonte para teste dos motores</i>	<i>11</i>
4.3.2	<i>Implementação física do componente Atuador</i>	<i>14</i>
4.4	INTERFACE DE PROGRAMAÇÃO DO MICROCONTROLADOR	14
4.5	CONCLUSÃO	15
5	REFERÊNCIAS BIBLIOGRÁFICAS.....	16

1 APRESENTAÇÃO DESTE TRABALHO – TEXTO DO PROFESSOR DA DISCIPLINA

Este trabalho de Microcontroladores e Sistemas Digitais tem por objetivo fazer uma revisão do conteúdo conceitual visto na disciplina teórica, mas que, agora, deve ser aplicado em um projeto prático. Para expor os conhecimentos este trabalho está dividido em 2 grandes partes, a saber: Parte 1 – Conceitual e Parte 2 – Simulação e/ou Montagem de Bancada.

A Parte 1 – Conceitual envolve conhecimentos teóricos, como MCU sua arquitetura em blocos, pinagem ou PORT's, funcionamento e programação em Linguagem C/C++ e trechos em Linguagem Assembly. Grande parte destes conhecimentos foram pesquisados no trabalho em Equipe – TPE na disciplina teórica e que serão requisitados agora para aplicação prática. Os conhecimentos em linguagem C/C++ serão aplicados diretamente na escrita do código-fonte da aplicação a ser desenvolvida para o SE utilizando a interface do Arduino ou de outro MCU disponível. As estruturas de programação, a lógica algorítmica a ser implementada, os comandos da linguagem, código-fonte, código-objeto, compilação, execução e todos os conceitos relativos a esta prática de programação são também objeto de interesse deste trabalho, mas não serão revistos, supondo conhecimento prévio consolidado pelo discente.

Via de regra, os projetos de Sistemas Embarcados têm aplicação real e lidam com variáveis de ambiente, como temperatura, umidade, pressão, para citar algumas mais conhecidas e outras mais atuais e complexas como, geolocalização, aceleração, medidas de inércia, todas analógicas com variação contínua no tempo e que precisam ser convertidas para um valor digital. Decorre daí a necessidade de compreender a Conversão Analógico para Digital (resumida pelo acrônimo ADC) e a importância de dimensionar qual deve ser a resolução mínima necessária (em bits do MCU) para garantir a exatidão da aplicação que o usuário deseja. Por outro lado, é de se observar também os requisitos se for desejável realizar a operação de Conversão Digital para Analógico (DAC).

A saída do SE, em geral, é utilizada para controlar um dispositivo em função da leitura da variável de entrada, por ex., ligar uma bomba d'água para irrigação quando a umidade do solo atinge o nível considerado como crítico. No entanto, às vezes, pode ser desejável controlar a velocidade de rotação de um motor DC. Isto pode ser efetuado por Modulação por Largura de Pulso (PWM) que é outra função de controle da potência entregue na saída de um pino (ou PORT).

A Parte 2 – Simulação e/ou Montagem de Bancada: aplicação dos conhecimentos para desenvolver um Sistema Embarcado. Nesta parte será executada a prototipação do SE, passo-a-passo, pois cada sensor e atuador deve ser testado, calibrado e aferido individualmente, tanto em hardware como em código-fonte para execução do sistema.

O projeto elaborado pelo professor, concomitante como as aulas de laboratório, servirá de modelo para que os alunos entendam o processo de construção de um SE considerando todo seu ciclo de vida.

2 RESUMO EXECUTIVO DO PROJETO

O projeto elaborado trata-se de uma automatização de uma linha de produção industrial, fazendo o transporte de um determinado produto, com o uso de um braço robótico que o leva até uma esteira e, posteriormente, encaminha o produto ao seu próximo setor.

O braço conta com dois servos motores base/garra, sendo que a base tem seu movimento em 90° ida/volta, e a garra foi programada para fechar uma angulação necessária para agarrar o objeto. A esteira trabalha no movimento cíclico de ida/volta.

O controle do projeto é feito por uma placa micro controladora Raspberry Pi 3 B+, que por sua vez foi programada em linguagem Python com a função de comandar a atuação dos servos motores e do motor DC da esteira, sendo este último, auxiliado por uma ponte H para realizar o controle de sentido da rotação do motor DC, que é basicamente um inversor de polaridade da corrente.

O uso de equipamento traz benefícios financeiros, por substituir mão de obra humana, que poderia desenvolver doença ocupacional devida a alta quantidade de repetição de um mesmo movimento; também traz benefícios para o controle de produção, pois ocorrem menos erros.

3 PARTE 1 – CONCEITOS ESSENCIAIS PARA PROJETO COM MCU

3.1 Componentes da Arquitetura de um Projeto com Microcontrolador

O projeto com Microcontrolador apresenta os componentes da Figura 1 abaixo que serão descritos individualmente a seguir.

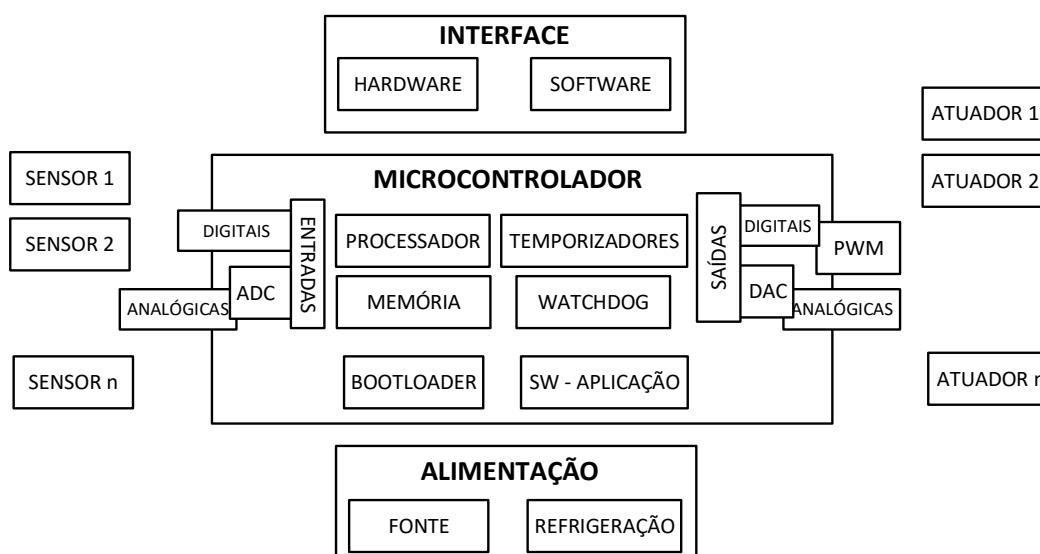


Figura 1 - Componentes essenciais de um Projeto com Microcontrolador

(Fonte: Material didático da disciplina. Autor: Prof. Dr. Jesús Franco Bueno)

3.2 Descrição conceitual dos componentes da Arquitetura do Projeto

Descreveremos a seguir, de maneira sucinta, os componentes e ferramentas que utilizamos para desenvolver o projeto.

3.2.1 Interface de usuário

Para realizarmos o gerenciamento das tarefas realizadas pelo projeto, desenvolvemos um algoritmo escrito na linguagem de programação *Python*, através de seu próprio interpretador, entretanto, por questões de interface, utilizamos também o software *Visual Studio Code* desenvolvida pela *Microsoft*, sendo ambas ferramentas gratuitas.

3.2.2 Microcontrolador escolhido

Devido ao fato de sua alta capacidade de processamento, familiarização com diversos componentes e sua alta proximidade com a linguagem *Python*, decidimos por escolher a *Raspberry Pi* modelo B *Plus*, visto que, além dos motivos anteriormente citados, levamos em consideração a ascensão de projetos desenvolvidos utilizando esse mesmo microcontrolador.

3.2.2.1 Entradas/saídas do Microcontrolador

O RPi é constituído por 40 pinos, sendo 28 pinos General purpose Input/Output (GPIO), 2 pinos 3.3V, 2 pinos 5V e 8 pinos de aterramento (ground), porta MIPI DSI display, Porta MIPI CSI câmera, 4 portas USB, entrada para cartão MicroSD, porta de saída HDMI, entrada de tensão sendo 5V/2.5A DC via conector USB e 5VDC via GPIO header e porta de saída PoE (Power over Ethernet) que permite usar o cabo Ethernet como fonte de energia. A figura 2 especifica os pinos GPIO disponíveis na Raspberry Pi 3 Model B+ e detalha o tipo da porta de cada pino.

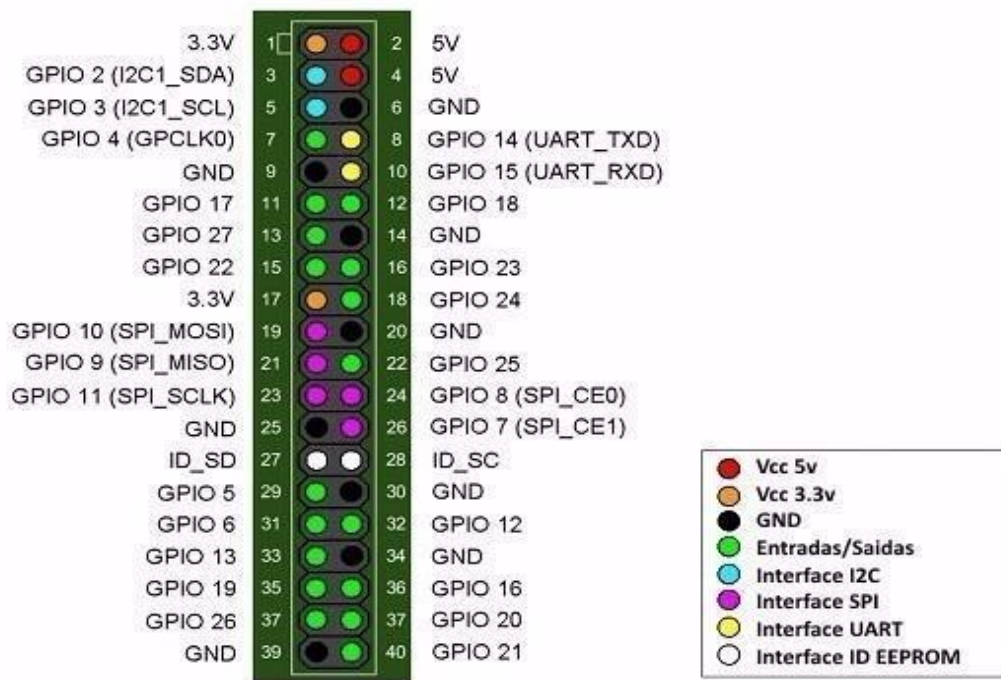


Figura 2 – Especificação de pinos GPIO disponíveis na Raspberry Pi

(Fonte: Controlando Raspberry Pi com Protocolo SSH. Autor: Rafael Fernando Bezerra)

3.2.3 Sensor

Nesta primeira versão do projeto, não utilizamos sensores, entretanto, aprimoraremos todas as etapas realizadas através da otimização do algoritmo, reconhecimento de objeto (produto) utilizando a biblioteca *OpenCV* e inclusão de sensores para realizar a contagem de objetos (produto)

3.2.4 Atuador

O projeto tem como atuadores o motor DC e o servo motor, sendo que, o gerenciamento de ambos é realizado através do algoritmo escrito em *Python*, conectado através de uma variável ambiente do *Windows*, responsável por especificar o caminho onde será realizado a troca de informações remotamente entre o servidor (notebook) e o cliente (Raspberry Pi), através do endereço de ip assumido pela placa. Em outras palavras, gerenciamos os pinos GPIO através dessa conexão especificada anteriormente. A parte física do motor DC utilizada na esteira é controlada através de uma ponte H, que é responsável por inverter a polaridade da tensão, sendo possível trabalhar em dois sentidos diferentes. Por fim, os servos motores utilizados no braço robótico são gerenciados através de sinais PWM (*Pulse Width Modulation*), ou seja, utilizamos o conceito *duty-cycle* para realizar a abertura e fechamento do ângulo necessário para a realização das etapas.

3.2.5 Alimentação e Refrigeração

A alimentação do projeto é realizada através do pino GPIO 5 Volts, este responsável por energizar os servos motores e por uma bateria externa de 12 Volts, utilizada para energizar o motor DC.

4 PARTE 2 – SIMULAÇÃO E/OU MONTAGEM DO PROJETO COM MCU

4.1 Descrição do projeto escolhido para execução com MCU

Processo de automatização de uma linha de produção industrial realizando o transporte de objeto de uma base ate uma esteira, utilizando um braço com pinça. Fazendo o controle do motor da esteira e dos dois servos do braço.

4.2 O Microcontrolador do Projeto

Raspberry Pi 3 B+

4.2.1 Implementação da Alimentação e Refrigeração

Conforme mencionado em tópicos anteriores, utilizamos dois servos motores para movimentação do braço robótico e um motor DC para manipular a esteira, juntamente com uma ponte H. Nas figuras 3, 4 e 5, é possível verificar a ilustração desses componentes.



Figura 3 – Componente físico do atuador servo motor 5V



Figura 4 – Componente físico do atuador motor DC 12V



Figura 5 – Componente driver motor ponte h l298n

4.3 Prototipação para teste, calibração e aferição do sensor

A prototipação do projeto, foi realizada através de testes realizados diretamente entre o algoritmo em *Python* e a placa *Raspberry Pi*, manipulando tarefas sequenciais do projeto. Através da prototipação, observamos vários pontos a serem ajustados, bem como, a angulação exata para fixar o objeto no braço mecânico, intervalos de tempo para realizar a inversão de polaridade do motor DC, definir a velocidade de abertura do servo motor localizado na base do braço mecânico, entre outros.

4.3.1 Esquemático para conexão do sensor FSR

A Figura 6 6 abaixo mostra a montagem do circuito utilizando Protoboard para conexão com o MCU e alimentação através de fonte externa.

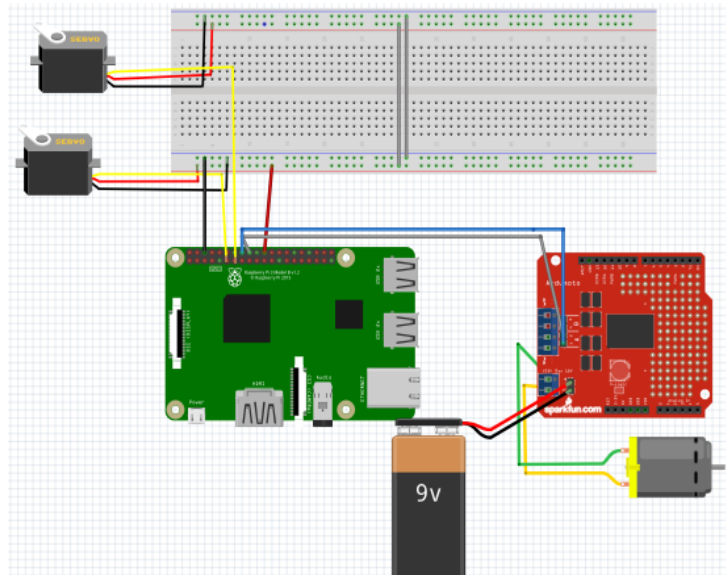


Figura 6 – Montagem do esquema para programação e teste dos motores

4.3.1.1 Código-fonte para teste dos motores

O algoritmo utilizado no projeto foi estruturado por módulos, ou seja, desenvolvemos para cada processo, uma biblioteca exclusiva para realizar a tarefa exigida.

```

/*****
# BIBLIOTECA DE VOZ (lib_voz.py)
import pyttsx3
import time

def voz_verif_obj():
    voz_init = pyttsx3.init('sapi5')
    voz_init.say('OBJETO DETECTADO!')
    voz_init.runAndWait()
def voz_deixa_obj():
    voz_init = pyttsx3.init('sapi5')
    voz_init.say('ACIONANDO A ESTEIRA!')
    voz_init.runAndWait()
def voz_ret_obj():
    voz_init = pyttsx3.init('sapi5')
    voz_init.say('RETIRE O OBJETO EM ATÉ 5 SEGUNDOS!')
    voz_init.runAndWait()
def voz_ret_est():
    voz_init = pyttsx3.init('sapi5')

```

```
    voz_init.say('RETORNANDO A ESTEIRA A POSIÇÃO INICIAL!')
    voz_init.runAndWait()
def voz_ret_braco():
    voz_init = pyttsx3.init('sapi5')
    voz_init.say('RETORNANDO O BRAÇO A POSIÇÃO INICIAL!')
    voz_init.runAndWait()

# BIBLIOTECA BASE BRAÇO MECÂNICO (lib_servo.py)
from gpiozero import AngularServo
import time
import os

os.environ["PIGPIO_ADDR"] = "192.168.43.156"
os.environ["GPIOZERO_PIN_FACTORY"] = "pigpio"

def sentido_positivo():

    srv_base = AngularServo(2, initial_angle = 0, min_angle=0, max_angle=90)
    srv_base.min()
    srv_base.angle = 0

    i=0.0
    while i<=90:
        srv_base.angle = i
        i = i + 1

def sentido_negativo():

    srv_base = AngularServo(2, initial_angle = 0, min_angle=-90, max_angle=0)
    srv_base.angle = 0

    i=0.0
    while i>=-90:
        srv_base.angle = i
        i = i - 1

# BIBLIOTECA ESTEIRA (lib_motor.py)
from gpiozero import Motor
import os
import sys
import time

os.environ["PIGPIO_ADDR"] = "192.168.43.156"
os.environ["GPIOZERO_PIN_FACTORY"] = "pigpio"
motor = Motor(26,20)

def esteira_positivo():

    for i in range (1,13):
        time.sleep(1)
        motor.forward(speed = 1)
```

```
motor.stop()

def esteira_negativo():
    for i in range (1,13):
        time.sleep(1)
        motor.backward(speed = 1)
    motor.stop()

# BIBLIOTECA GARRA BRAÇO MECÂNICO (lib_garra.py)
from gpiozero import AngularServo
import time
import os

os.environ["PIGPIO_ADDR"] = "192.168.43.156"
os.environ["GPIOZERO_PIN_FACTORY"] = "pigpio"

def abrir_garra():

    srv = AngularServo(4,initial_angle = 0, min_angle=0, max_angle=90)
    srv.angle = 0
    time.sleep(1)

def fechar_garra():

    srv = AngularServo(4,initial_angle = 0, min_angle= -90, max_angle=0)
    srv.angle = -3
    time.sleep(1)

#PROGRAMA PRINCIPAL
import lib_servo
import lib_voz
import lib_motor
import lib_garra
import os
import time
import sys

lib_garra.abrir_garra()
time.sleep(2)
lib_garra.fechar_garra()
time.sleep(2)
lib_servo.sentido_positivo()
time.sleep(2)
lib_garra.abrir_garra()
time.sleep(2)
lib_voz.voz_deixa_obj()
lib_motor.esteira_positivo()
lib_voz.voz_ret_obj()
time.sleep(5)
lib_voz.voz_ret_est()
```

```
lib_motor.esteira_negativo()  
lib_voz.voz_ret_braco()  
lib_servo.sentido_negativo()
```

4.3.2 Implementação física do componente Atuador

Ponte H

Tem como principal função o controle de velocidade e sentido de motores DC escovados, podendo também ser usado para controle da saída de um gerador DC ou como inversor monofásico.

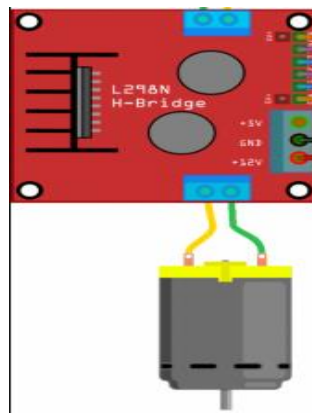


Figura 7 – Componente físico do atuador com motor

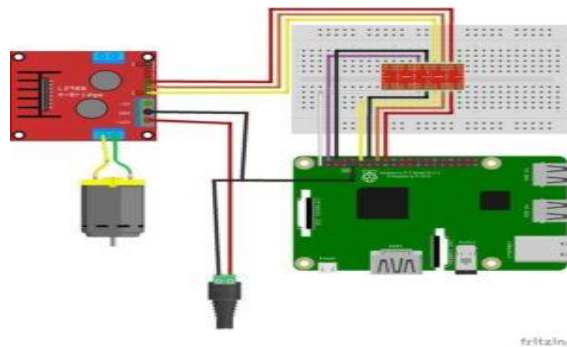


Figura 8 – Montagem do esquema para programação e teste do atuador

4.4 Interface de Programação do Microcontrolador

A figura 9, representa o ambiente de programação *Python*, juntamente com seu interpretador, utilizado durante todo o desenvolvimento do projeto.

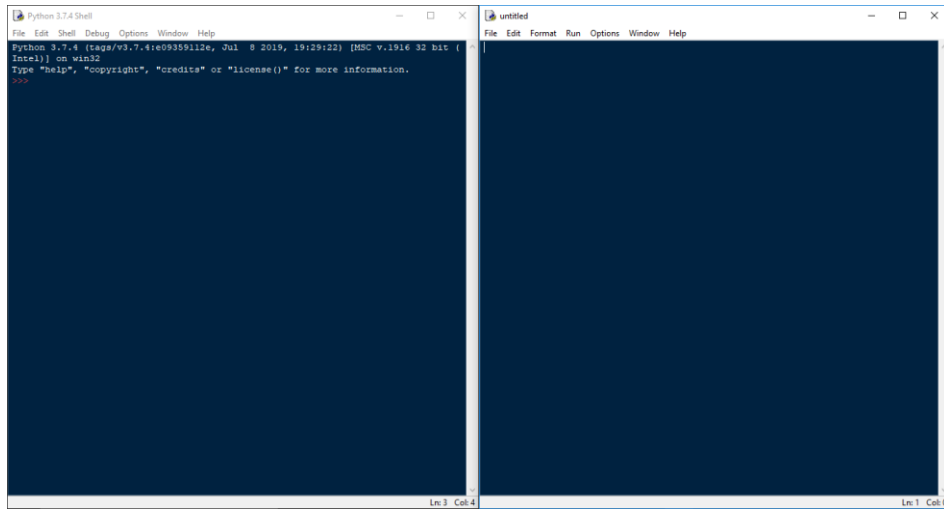


Figura 9 – Interface de Programação para Microcontrolador - IDLE Python

4.5 Conclusão

Através desse projeto, conseguimos aplicar os conceitos ministrados durante a disciplina de Microcontroladores e Sistemas Digitais sobre componentes, entradas e saídas analógicas e digitais, modulação por largura de pulso, entre outros tópicos presentes na ementa ministrados durante o semestre. Acrescentamos também, conhecimentos em ferramentas que não tivemos contato anteriormente, bem como, a linguagem de programação *Python*, o *software Visual Studio Code*, porém, conseguimos aplicar conceitos fundamentais já lecionados em disciplinas anteriores como Circuitos Elétricos e Eletrônica Analógica e Digital, pois nos auxiliou nas montagens do esquemático do projeto e voltagens de componentes utilizados.

5 REFERÊNCIAS BIBLIOGRÁFICAS

Haeder, Adam. Alta Books, ed. Certificação Linux LPI: rápido e prático, 2012 3 ed. Rio de Janeiro: [s.n] 520 páginas.

Raspberry Pi, F. (2019). Datasheet Raspberry Pi Compute Module 3+. [online] Raspberrypi.org. Disponível em: https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM3plus_1p0.pdf [Acessado em: 23 Ago. 2019].

Raspberry Pi, F. (2019). Raspberry Pi Foundation - About Us. [online] Raspberry Pi. Disponível em: <https://www.raspberrypi.org/about/> [Acessado em: 23 Ago. 2019].