|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MEMORANDUM** | | |  | |
|  |  | | | |
| **To:** | Charlie Refvem, Lecturer, Department of Mechanical Engineering, Cal Poly SLO | | | |
|  | [**crefvem@calpoly.edu**](mailto:crefvem@calpoly.edu) | | | |
| **From:** | Michael Shokoohi | | |
|  | Msshokoo@calpoly.edu | | |
| **Date:** | 10/2/2025 |
| **RE:** | **ME 405-01 Mecha-02** |
|  |  | | | |

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

|  |
| --- |
|  |
|  |
| Figure 1. This figure shows the charging curve for the RC circuit used in this lab. The Measured charging curve was plotted using ADC data from the nucleo. The Theoretical curve was plotted using a derived time constant from component values and equation [1]. [1] |

|  |
| --- |
|  |
|  |
| Figure 2. The figure shows the linearized data which is plotted using the ADC data and equation [2].[1] The trendline shown was made using the first 200 values in the data set to capture the slope of the linear region. The slope of -11.755 corresponds to a time constant of .0851 |

When comparing the time constant derived from the linearized charging curve .0851 with the theoretical time constant of .0749 we find that there is an 11.97% Error. This is to be expected because the components themselves have a tolerance between 5%-10%.

|  |  |  |
| --- | --- | --- |
| [1] | Charlie Refvem, Lab 0x00 Notes. | |
|  | |
|  | |
|  | |

**Attachments (1)**

from pyb import ADC, Pin,Timer,ExtInt

from array import array

import time

"""

Author: Michael Shokoohi

Term: Fall 2025

Course: Mechatronics ME 405

Assignment Description:

Notes: Resistor: 20.13 k Ohm

       Capacitor: 3.72 uF

       Resistor to Nucleo input: 6.10 k Ohm

       Signal Goes to PC0

       Step output goes to PC1

"""

def tim\_cb(tim):

    '''

    Call back func that triggers input to RC circut and collects data for 5X as long  as

    Tau.

    Trigger output on call back

    after that never drop it

    record for 5Tau

    append data to array

    '''

    global data

    global i

    #Activating the step response on second run through callback function.

    if i==1:

        PC1.high()

    if i<1000:

        data[i]= adc.read()

        i=i+1

    else:

        tim.callback(None)

def Publish():

    '''

    loop through the data array and push it to the putty terminal

    '''

    global data

    global i

    idx=0

    print(data)

    for idx, value in enumerate(data):

        print(f"{idx}, {value}")

    data= 1000\*[0]

    i=0

def ExecuteSequence():

      # Assign the callback function

    tim7.callback(tim\_cb)

    time.sleep\_ms(1000)

    #tim7.callback(None)     # disable the callback

    PC1.low()

    Publish()

def FlipFlag(\_):

    global Execute

    if Execute:

        Execute=False

    else:

        Execute=True

if \_\_name\_\_=='\_\_main\_\_':

    #Pre allocating array for data storage

    # H is used to represent data type sint..

    data = array('H', 1000\*[0])

    Execute=False

    #Creating timer object for Timer number 7

    tim7 = Timer(7, freq=1000)

    #Config PC1 as digital output (Step input)

    PC1 = Pin(Pin.cpu.C1, mode=Pin.OUT\_PP)

    #Config PC0 as Analog pin

    PC0= Pin(Pin.cpu.C0, mode=Pin.ANALOG)

    #Config ADC to be attached to PC0

    adc= ADC(PC0)

    i=0

    # Config button to start program

    button\_int = ExtInt(Pin.cpu.C13, ExtInt.IRQ\_FALLING,

                    Pin.PULL\_NONE, FlipFlag)

    while(True):

        if Execute:

            ExecuteSequence()

            FlipFlag('a')