Journal of Electrical Engineering, Electronics, Control and Computer Science
JEEECCS, Volume 2, Issue 5, pages 21-24, 2016

Micromeasure System through Electrical Palpation - Interfaced with the User by a Touchscreen / Raspberry Computer

Gheorghe Andrei COSMIN, Emil DIACONU
Valahia University of Târgoviște, Faculty of Electronics and Telecommunications
E-mail: gheorghe_andrei89@yahoo.com

***Abstract* - This papers shows how a positioning robot can work via a touchscreen display connected with a Raspberry type computer. The application consists of a PWM signal manipulating from a cursor interface that includes two linear graphs, whose scroll bars change specific parameters and fill factor. Finally, the signal is amplified by ordering a electromagnetic actuator through which a work device(a peak) is moved in a controlled linear space.**

Key-Words: efficient, measuring system, Raspberry

I. INTRODUCTION

Microrobotics and robotic applications are generated by standard methods of work that includ notions such as order processing , control and interface. Operating interfaces used by operators include specific elements for robotic equipment operations and enables the management of the workflow. The operator inputs the data, then visualizes the process state and intervenes to adjust by comumnicating vitually through the interface with the machine. Micromecatronics refers to the technical process of automatization of aplications developed in micrometer areas[1]. Specific actuators are ordered on different physical principles, notable among them is a special class, the electromagnetic actuators. In the patent[2] is described an electromagnetic actuator that has the capacity to develop micrometer displacements by energizing a coil generating a magnetic force acting a elastic mechanical arm. The electrical signal is PWM controled[3], which allows control over the energizing coil. PWM current hardware units include functions that can be acceseed through varios code sequences for a result it was chosen for use a Raspberry Pi-type proccesing unit[4]. The touchscreen display was chosen for a working adjacent hardware unit. The programming used requires a programming language adapted[5], in this case Python. The program interface provides good handling, offering its very intuitive graphics-type linear slider.

II. THE PRINCIPLE OF MEASURING AND THE PROCESS MANAGEMENT OF THE PROGAM'S ALGORITHM

In Figure 1 is shown the measuing principale which consists of:

Vertical micrometer displacement of the tip (1). The maximum range of movement representing measurement range (Im).

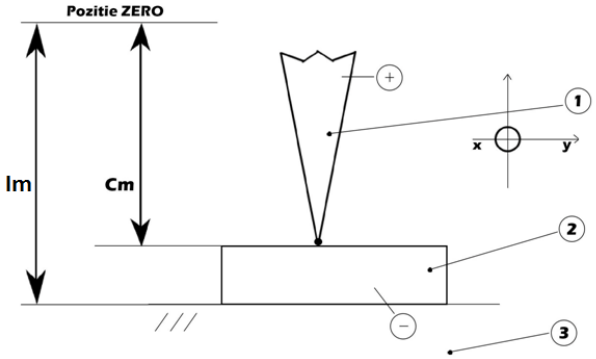


Figure 1. Measuring principle

Measurement consists of travel withholding when a contact is made between the tip(1) and part (2) located on the support (3). The contact timing accuracy is done using an electrical contact, intercepted by the machine.

III. PRESENTING THE EXPERIMENTEL STAND

In Figure 2 are shown the basics parts that are needed for the completion of the experimental part.

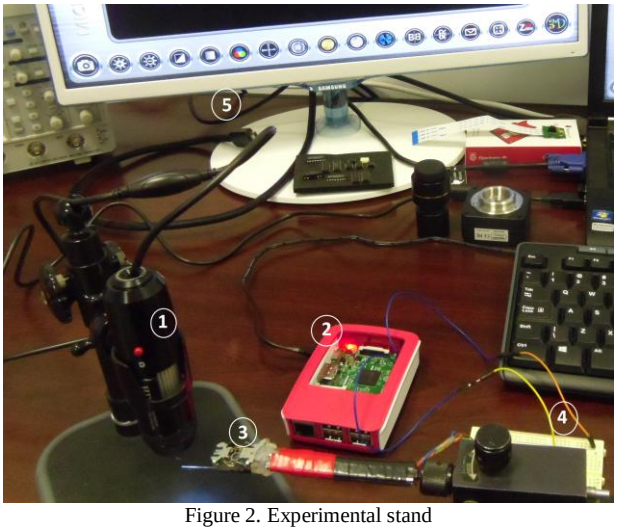


Figure 2. Experimental stand

The videomicroscop (1) is used for intercepting movements of the working device driven by the electromagnetic actuator (3). Signal processing and interpretation of the control orders that are received from the touchscreen interface (6 from Figure 3) is performed by the Raspberry unit (2).

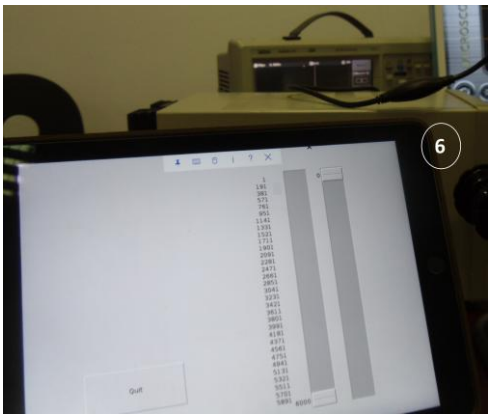


Figure 3. Experimental stand (touchscreen interface)

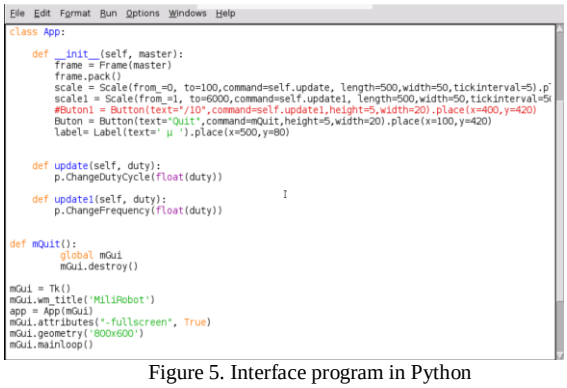
Screen (5) is used for viewing the workspace by the operator who develops micrometric movements of the working device (7 in Figure 4).

Figure 4. Eperimental stand (working device)

The working device (7) is simulated in this work development application interface. By means of a glass tube with a diameter of 600 micrometers.
The touchscreen display is basically a tablet using the Raspberry unit interconnected via local Wi-Fi.

IV. THE PROGRAM MADE FOR INTERFACE OPERATION

The program includes two variables (scale, scale 1), which refers to the specific parameters of the PWM signal, namely frequency and duty cycle.

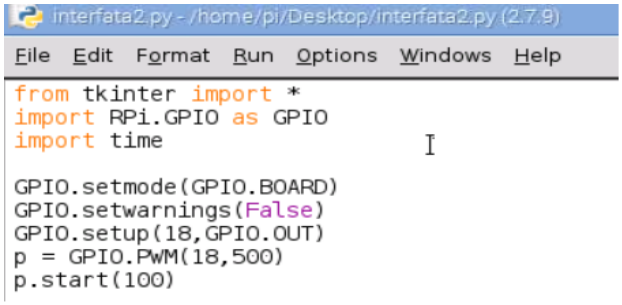


```
File  Edit  Format  Run  Options  Windows  Help
class App:

    def __init__(self, master):
        frame = Frame(master)
        frame.pack()
        scale = Scale(from_=0, to=100,command=self.update, length=500,width=50,tickinterval=5).p
        scale1 = Scale(from_=1, to=6000,command=self.update1, length=500,width=50,tickinterval=5
        #Buton1 = Button(text="/10",command=self.update1,height=5,width=20).place(x=400,y=420)
        Buton = Button(text="Quit",command=mQuit,height=5,width=20).place(x=100,y=420)
        label= Label(text=' µ ').place(x=500,y=80)


    def update(self, duty):
        p.ChangeDutyCycle(float(duty))

    def update1(self, duty):
        p.ChangeFrequency(float(duty))


def mQuit():
        global mGui
        mGui.destroy()

mGui = Tk()
mGui.wm_title('MiliRobot')
app = App(mGui)
mGui.attributes("-fullscreen", True)
mGui.geometry('800x600')
mGui.mainloop()
```

Figure 5. Interface program in Python

In Figure 5 shows the program, and in Figure 8 is the interface generated from running the program.

Breaking down the program:



```
interfata2.py - /home/pi/Desktop/interfata2.py (2.7.9)
File  Edit  Format  Run  Options  Windows  Help
from tkinter import *
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(18,GPIO.OUT)
p = GPIO.PWM(18,500)
p.start(100)
```

Figure 6. Startup Code

In figure 6 shows the main code lines that are used for the interface, we can see at the begining of the program that are imported from the Python library 3 main functions:

1) Tkinter – the library that is responsable for the grafical interface.
2) RPI.GPIO – the library that is responsable for controling the GPIO pins on the Raspberry bord.
3) Time – the library that is used for starting and stoping the aplication at a certain time.

The next functions such as „GPIO.setmode" is used for setting up the board pins in a predefined order, the „GPIO.setup" is used to select the only PWM compatible pin on the board which is pin 18.

In Figure 7 we can see where the code for the construction of the interface such as location,size defined by „scale" , „scale1" and „Buton" and the function of the input buttons/sliders.

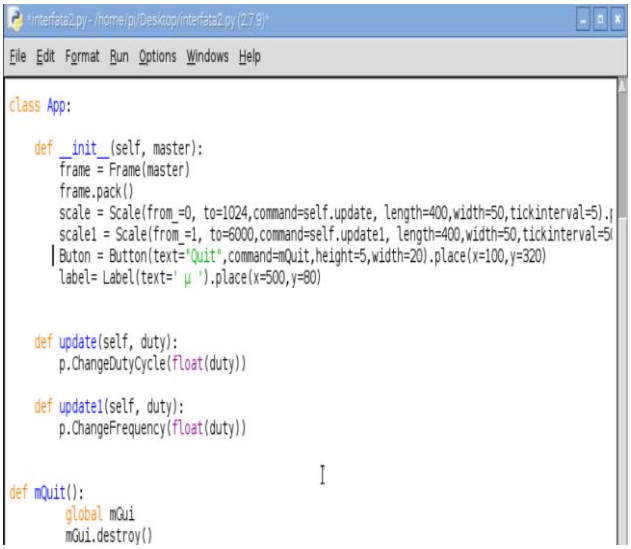


Figure 7. Aplication code

The main liniar sliders that are used in the interface are controling the frequency and duty cycle.

1) ChangeDutyCycle – is the comand that is self updating when you move the slider from 0% to 100%.
2) ChangeFrequency – is the comand that is self updating when you move the slider from 1 Hz to 6000 Hz.

The „mQuit()" function lets the user exit the program.
In this interface were used as input two liniar graphics sliders.

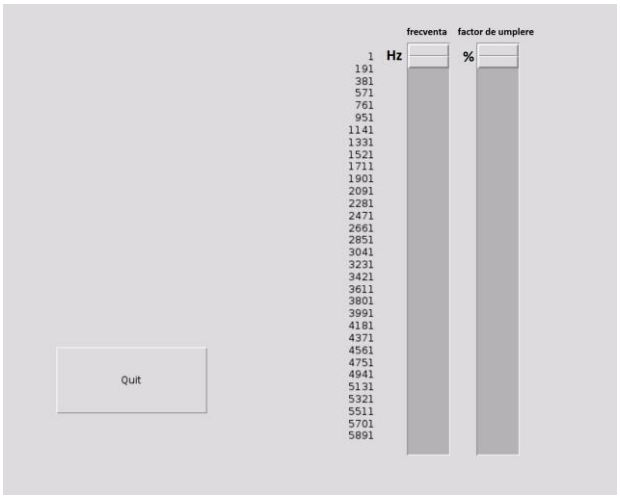


Figure 8. Interface

The rememberd parameters are injected into the standard PWM function of the Raspeberry Pi hardware, more exactly the GPIO pin nr.18. For exiting, it was introduced and exit button, the operator can exit the application at any time.

V. CONCLUSIONS AND PICTURES FROM THE EXPERIMENT

Trough this paper we demonstrated the practical connectivity for a touchscreen graphical user interface on a signal generator device and high maneuverability of this type of interface through which a user can control a micomecatronic device.

Connectivity refers to any type of touchscreen device that supports Wi-Fi connection, such as a mobile phone or tablet.
As a perspective, usage of a specific program for videomeasure will work side by side with the system set up for the PWM control signal for the movement of the micrometric actuator will be needed.

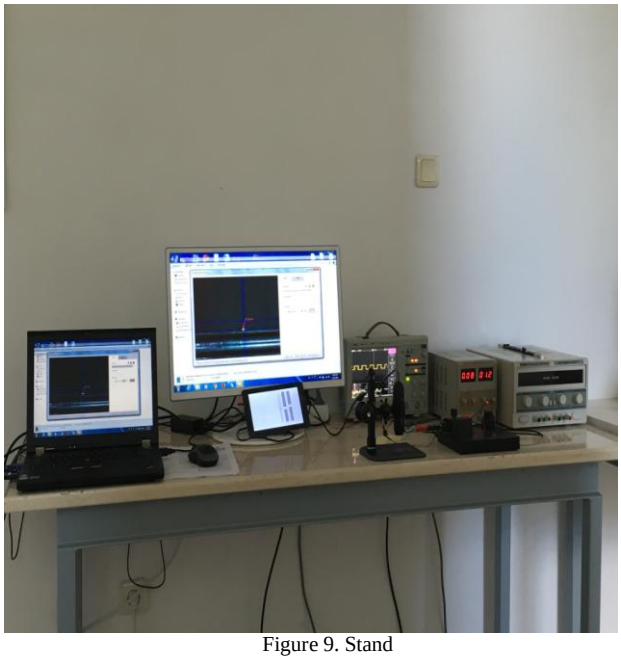


Figure 9. Stand

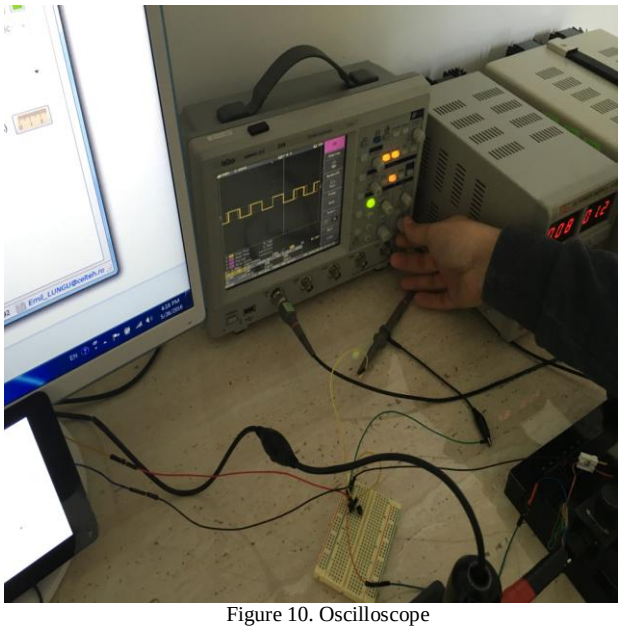


Figure 10. Oscilloscope

Figure 11. Holding Bracket

Figure 12. Video Microscop

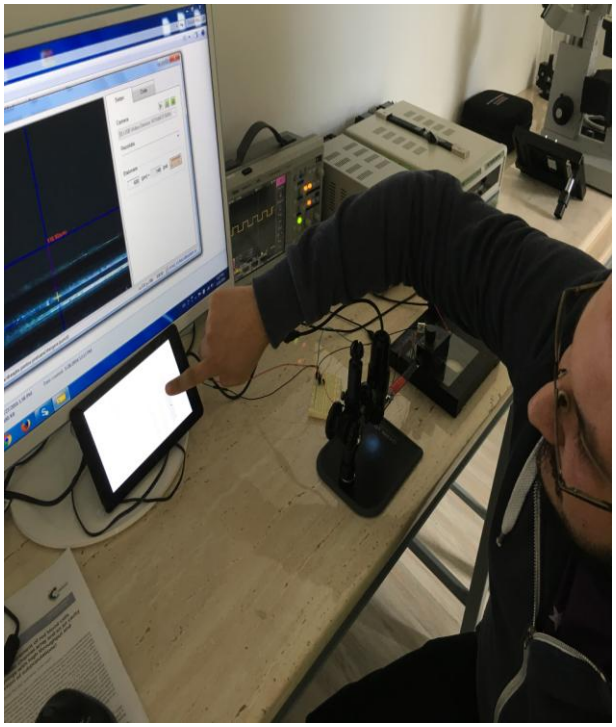


Figure 13. Touchscreen Panel

VI. REFERENCES

[1] Ardeleanu M. ș.a., Aplicații mecatronice și micromecatroince, București, Matrixrom, 2014
[2] Brevet RO 129160 A2, ”Dispozitiv de actionare magneto-piezoelectric pentru micromanipulare”
[3] https://learn.sparkfun.com/tutorials/pulse-width-modulation
[4] https://www.raspberrypi.org/
[5] https://www.python.org/