Role of Python,
in Education and beyond

PyCon India 2021

Ajith Kumar B P

bpajith@gmail.com

Talking about ...

- How I came across Python (old story !!!)
- A Software superpower
  - Perceptions and reality
  - Open source contributions
- IT Training at School level
  - Present status
  - Desirable options
- Software as a supporting tool for core subjects
- Computer Programming for Everybody
  - Advantages of expanding the base

Pycon 2015 :https://www.youtube.com/watch?v=bDbXUAOIo4w

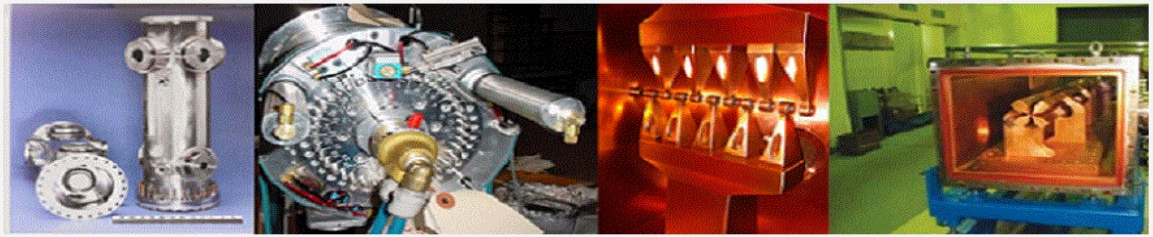
Inter-University Accelerator Centre, New Delhi

www.iuac.res.in/accel/index.html

IUAC

Inter - University Accelerator Center
विश्वविद्यालय अनुदान आयोग का एक स्वायत अनुसंधान केंद्र
An Autonomous Research Centre of University Grants Commission
नई दिल्ली New Delhi

ORGANIZATION | ACCELERATORS | RESEARCH | FACILITIES | STUDENTS CORNER | USER INFO | DOWNLOAD | DASHBOARD



Pelletron

The 15 UD pelletron is a versatile, heavy ion tandem type of electrostatic accelerator. In this machine, negative ions are produced and preaccelerated to ~300KeV in Ion Source and injected into strong Electrical field inside an accelerator tank filled with SF6 insulating gas.—more

Superconducting Linear Accelerator

The accelerating structure for the superconducting linac booster for the 15 UD Pelletron at IUAC is a Niobium Quarter Wave Resonator, designed and fabricated as a joint collaboration between IUAC and ANL, USA. Initial resonators required for the first linac module were fabricated at ANL.—more

Pelletron Accelerator RBS-AMS Systems

Pelletron Accelerator RBS-AMS Systems (PARAS). Rutherford Backscattering Spectrometry (RBS) facility with 1.7 Million Volt Pelletron accelerator has been installed at IUAC. The facility is equipped with 1)Alphatross ion source—more

Accelerator Mass Spectroscopy

Accelerator Mass Spectrometry with 15UD Pelletron at IUAC is first AMS facility in India for the detection of ^{10}Be and ^{26}Al.—more

Low Energy Ion Beam Facilities

The Low Energy Ion Beam Facility (LEIBF) at the Inter-University Accelerator Centre provides multiply charged ion beams at a wide range of energies (a few keV to about an MeV) for experiments in Atomic, Molecular and material sciences.—more

Ongoing Projects

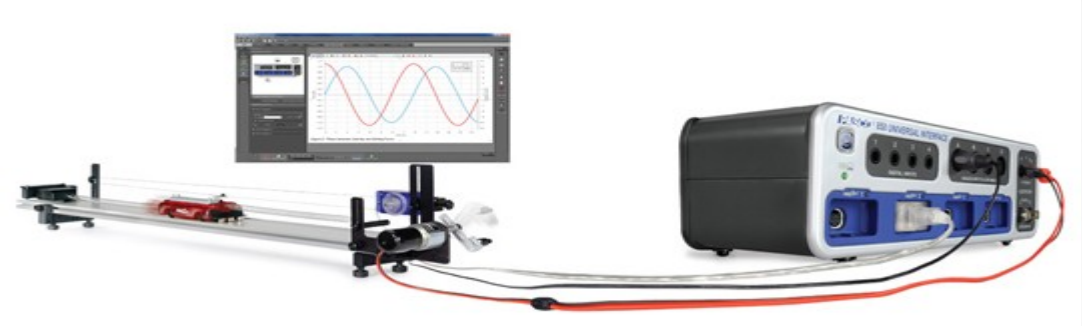
High Current Injector

The High Current Injector (HCI) project at Inter University Accelerator Centre would use a Radio Frequency Quadrupole (RFQ), Drift Tube Linac (DTL) and low beta superconducting cavities to accelerate heavy ions—more

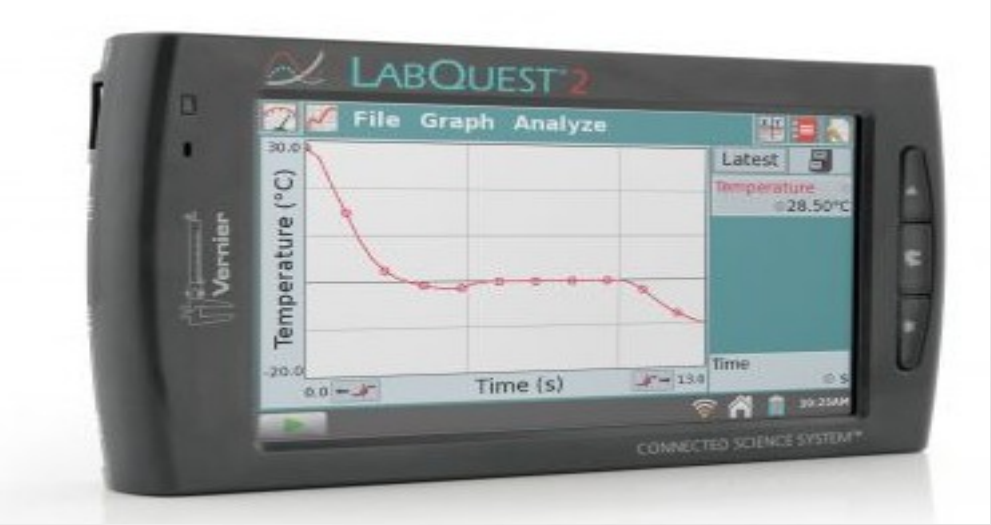
Free Electron Laser

Details will be available soon

A Problem: Lack of equipment for teaching Science



Expensive and proprietary solutions, not affordable for developing nations.



Solution: Develop cost-effective lab equipment

Sensor/Control Elements ↔ Signal Processing ↔ Display/ User Interface

Sensors. Consumer electronics items like microphones

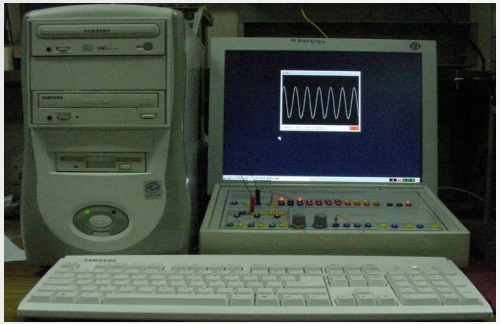
Signal processing and Interfacing Circuits

A laptop or Raspberry Pi (UI Software)

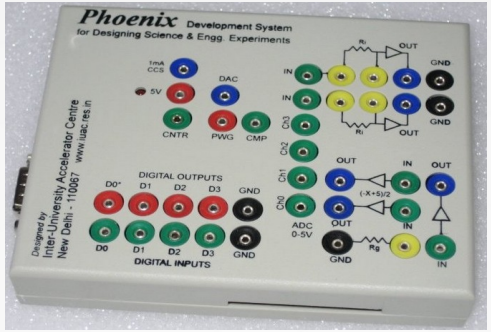
Designs were open sourced

Products from the PHOENIX project, started in 2005

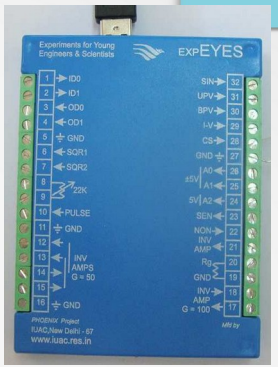
https://www.iuac.res.in/phoenix



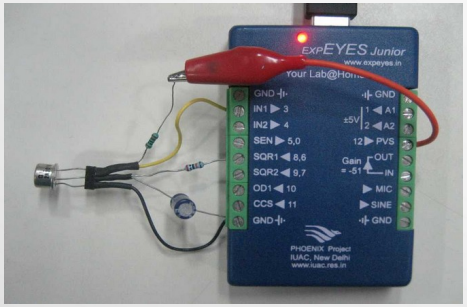
Parallel port version, 2005



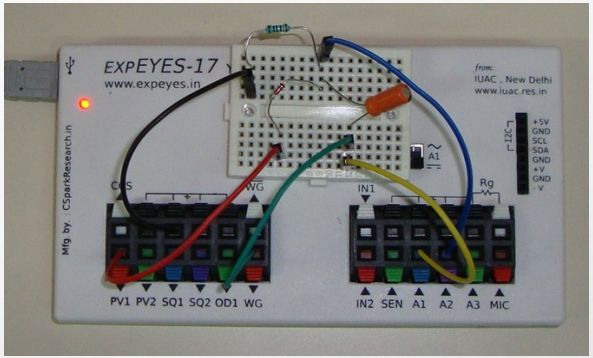
Serial port version (with uC), 2006



ExpEYES (USB), 2011



ExpEYES Junior, 2012



ExpEYES-17, 2017

What is expEYES meant for ..

Students : An affordable tool for doing experiments, anytime anywhere. Freedom from the lab timings.
(Relevant now, due to the pandemic)

Teachers : A tool for classroom demos, experiments and also to develop new experiments.

Engineers : An open system that combines basic physics, electronics, micro-controller programming, computer interfacing, GUI programming and scientific computation.

Hobbyists : A nice tool to kill more time with less money.

Collective effort, Inspired by the Free Software ideals ...

Pramode CE,
brought in Python (2005)

Georges Khaznadar,
Debian packaging (2005)

Praveen patil
GSoC 2014

Jithin,
ExpEYES-17 (2017)

Phoenix at Lycée Jean Bart, France

Feedback and support
from hundreds of
Physics teachers.

The Free Software movement

- Richard Stallman (Free as in Freedom)
- Physics BA, Harvard, 1974. Programmer at MIT
- In 1983, launched the GNU Project to write a completely free operating system.
- The Free Software Foundation, 1985 (FSF India, 2001)
- GNU General Public License
- By 1991, GNU OS was almost complete

  (except the Kernel, HURD)

https://www.gnu.org/ https://www.fsf.org/

The Linux Kernel, from Linus Torvalds

Computer science student at University of Helsinki, Finland.

25 August, 1991, he made a posting to the comp.os.minix newsgroup:

“Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since April, and is starting to get ready.”

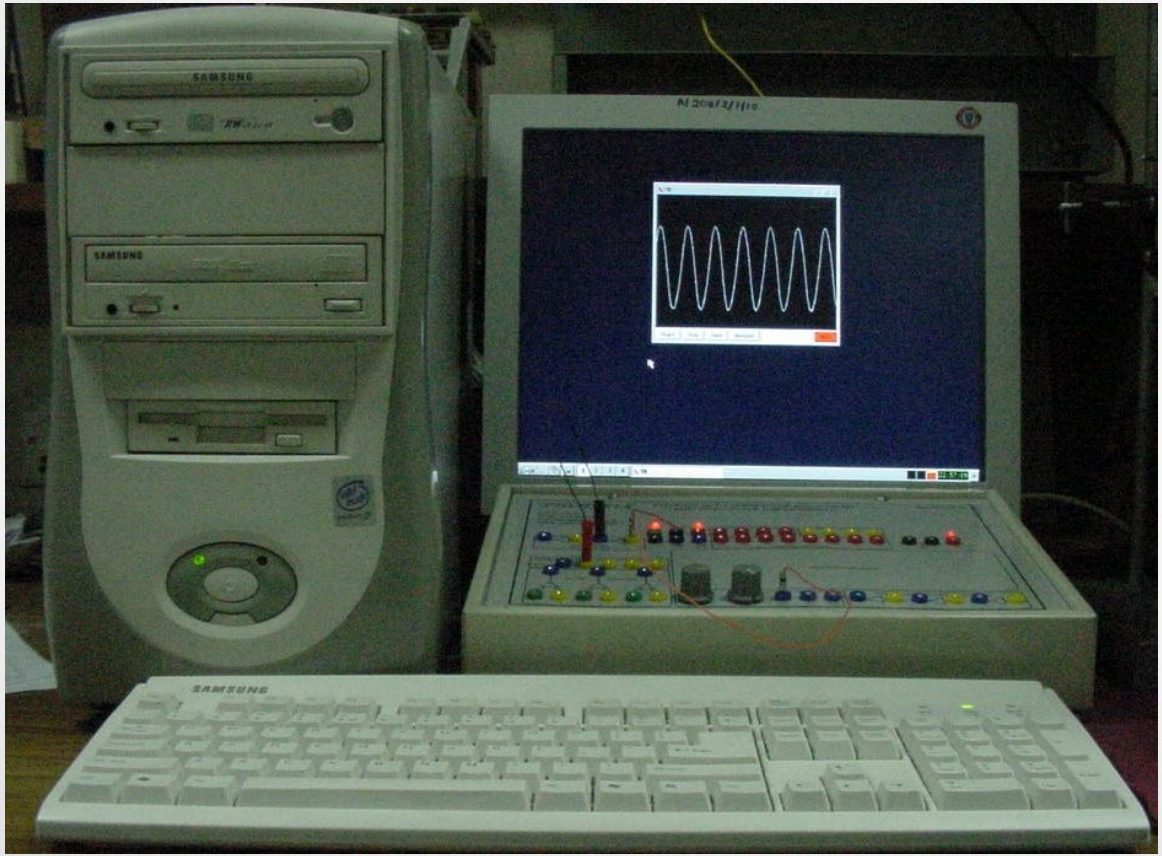
Version 0.99 was released in 1992 uder GNU GPL.

A complete OS , kernel version 1.0 plus the other parts from the GNU project, was released in March 1994.

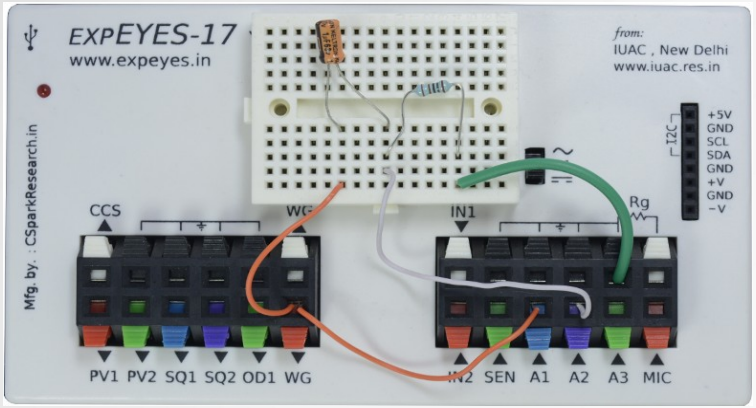
Search
“just for fun linus”

https://www.kernel.org/

Writing an Oscilloscope in “C” drove me to Python

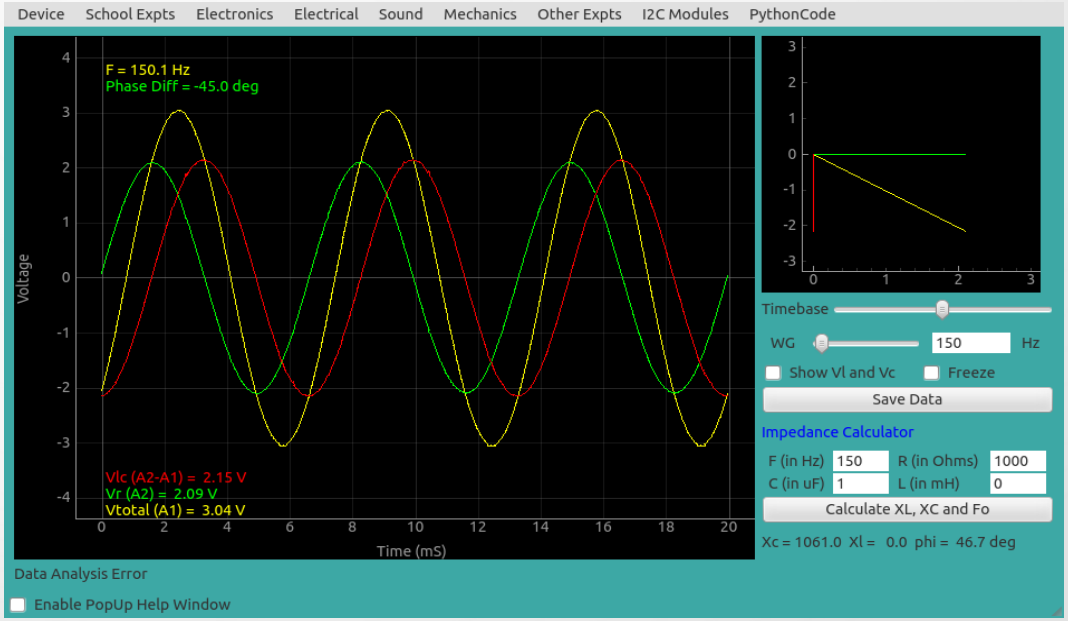


Affordable Scientific Equipment.. ExpEYES



www.expeyes.in

Python powered electronics Instrumentation.



Studying LCR circuits
Amplitude, frequency and phase of AC are measured.

There is no frequency or phase meter.

Done by fitting data with equations using Scipy package.

www.expeyes.in

https://expeyes.in

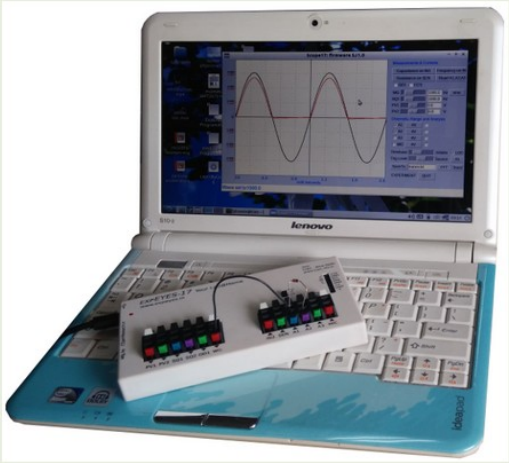
ExpEYES ... Your Lab@Home

Low Cost Science Experiments using Computers

ExpEyes

Experiments | Design | How to Buy | Software | Android | python | MicroHOPE | People | News

Experiments for Young Engineers and Scientists



- A tool for learning science by exploring and experimenting.
- Covers Experiments at Under Graduate and Graduate level
- 4 channel Oscilloscope, 1Msps, +/-16V input range
- Sine/Triangular Wave Generator, 5Hz to 5kHz
- Programmable voltage sources, +/5V and +/-3.3V
- Frequency Counter and time measurements.
- Supports I2C standard sensors
- 12bit analog resolution.
- Open Hardware and Free Software.
- Software in Python programming language.
- get started with the User Manual and Videos
- Tested on Raspberry Pi (deb file)

Teacher training at Inter-University Accelerator Centre

www.iuac.res.in/phoenix

Topics of Interest

Scientific Computation with Python

Experiments Using ExpEYES.

Open hardware & Software

A Python book

https://scischool.in/python/index.html

Home Resources Books

Python for Education

A Python book meant to support teaching science and mathematics

This is the revised version of the book. The First Version was written before ten years.

The changes are mainly due to Python3, chapters related to numerical methods are also revised.

Download the PDF version from HERE.

Download the Example Programs

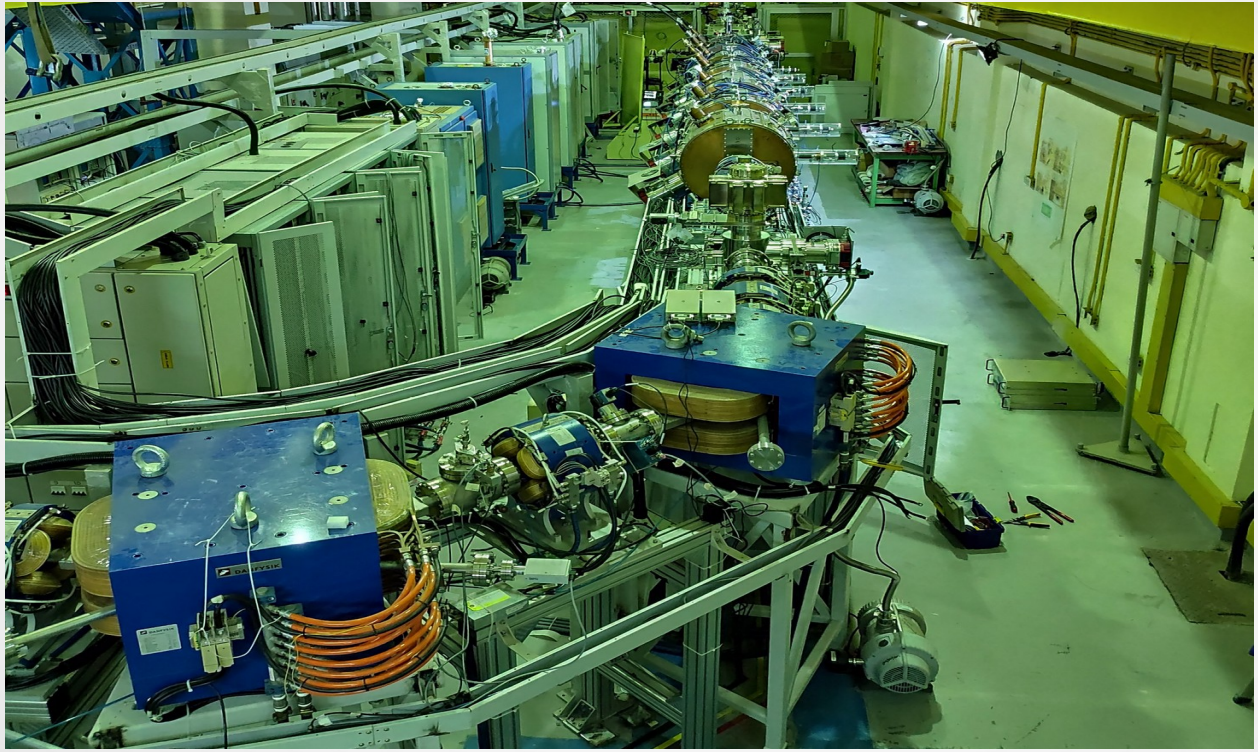
You can practice python coding from websites like Trinket.io. Copy the following lines

Download

```
from pylab import *
x = linspace(-pi, pi, 50)
plot(x, sin(x))
show()
```

Open Trinket.io , paste it and execute.

Controlling Particle Accelerators



```
Import pelcon
p = pelcon.pelcon()
Print p.get_voltage('CPS031', 'VC', 'KV', 2.2)
```

Any of the 2000 signals can be controlled/monitored from a Python front end.

Are we really doing well in Software ?

https://niti.gov.in/planningcommission.gov.in/docs/aboutus/taskforce/tk_know.pdf

Popular Software Applications

- Electronics and Mechanical CAD packages
- Weather forecast
- Simulations in Science and Engineering
- Language Compilers
- Operating Systems

Is there any indigenous product /development ?

https://www.youtube.com/watch?v=TlrzzeFsMXo

Coding as a hobby first...

Observation

Our IT education includes computer applications but programming is limited to a small number of students.

Why ?

Coding at school level. Will it work ?

My first interaction with 10th std. Students, in 1997

Basic concepts
using MS DOS+ windows 3.1
were taught.

Students were attracted to BASIC programming, even though they wrote only few lines.

Did we take the wrong route ?

Our school level IT training started pushing Office Package as the educational software.

- Struggling to plot graphs using Spreadsheets
- Making Powerless and Pointless presentations
- Focusing more on typesetting than content.

Motivation behind Raspberry Pi

..decline in the numbers and skills levels of the A Level students applying to read Computer Science (at Cambridge) in each academic year. From a situation in the 1990s where most of the kids applying were coming to interview as experienced hobbyist programmers, the landscape in the 2000s was very different; a typical applicant might only have done a little web design.

Something had changed the way kids were interacting with computers. A number of problems were identified: the colonisation of the ICT curriculum with lessons on using Word and Excel, or writing web pages;

https://elinux.org/RPi_General_History

IT as a Supporting Tool in Education

- Computation
- Simulation
- Data Visualization

Where does Python fit in these requirements ?

Learning Science & Mathematics

Can we use Python as a tool for that ?

How much effort is required ?

What are the benefits ?

- Easy to learn (not much to mug up)

```
#include <iostream.h>
void main()
{
cout << "Hello world" << endl;
}
```

```
print ("Hello world")
```

Other advantages of Python may be demonstrated better using some examples.

(Nothing against C, it is still my favorite language.
But it may not be the best option to start with)

A simple (but useful) application

```
for trial in range(1,11):
    print ('%3d x 5 = %3d'%(trial, trial * 5))
```

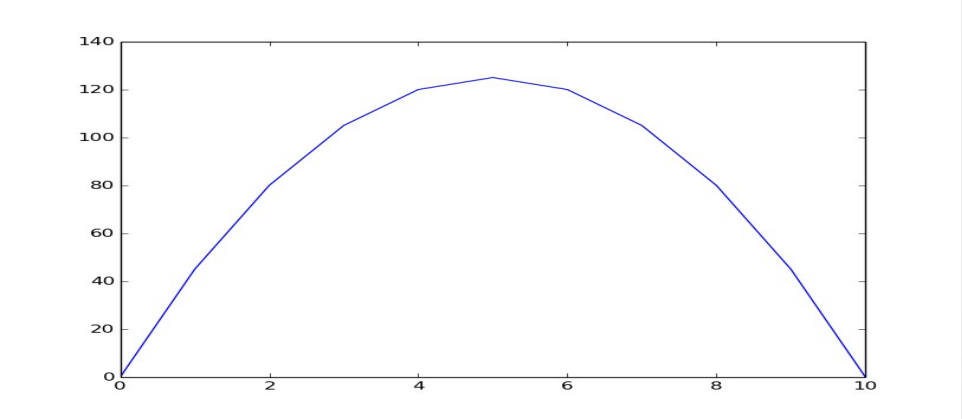
```
  1 x 5 =   5
  2 x 5 =  10
  3 x 5 =  15
  4 x 5 =  20
  5 x 5 =  25
  6 x 5 =  30
  7 x 5 =  35
  8 x 5 =  40
  9 x 5 =  45
 10 x 5 =  50
```

“Learning to Program”
Vs
“Programming to Learn”

School Physics

$$x = x_0 + ut + \frac{1}{2} a t^2$$

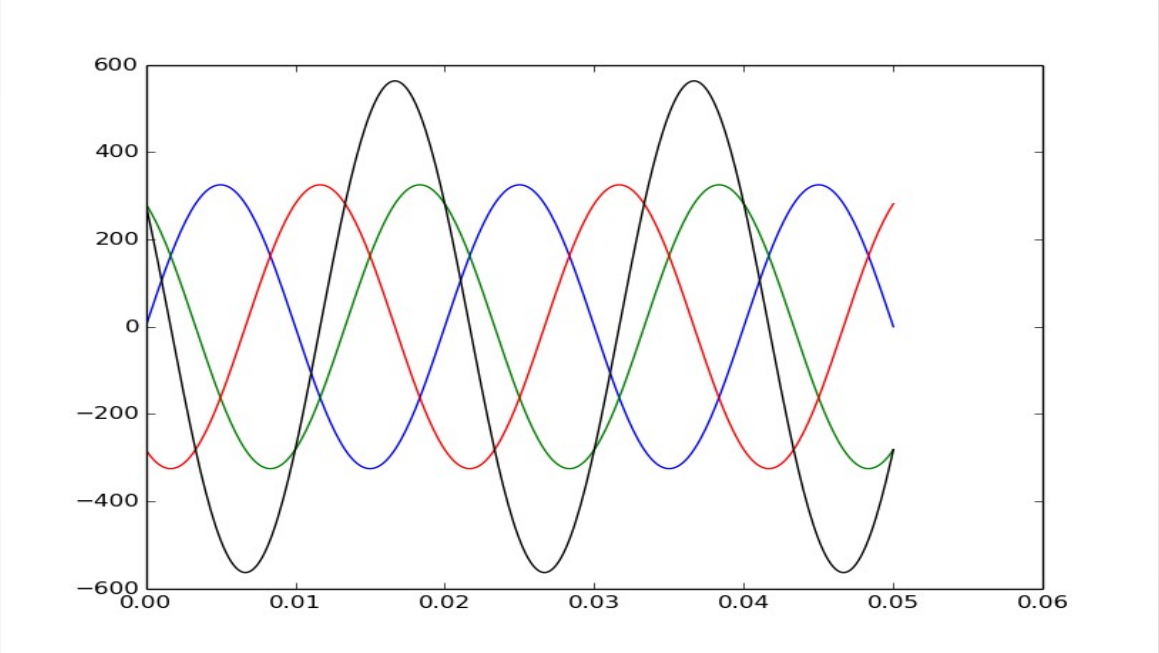
```
from pylab import  *
t = linspace(0,10,11)

x0 = 0.0        # initial position
u  = 50.0       # initial velocity
a = -10         # negative accn

x = x0 + u*t + 0.5 * a * t**2
plot(t,x)
show()
```



Time – Displacement plot

Making graphs to explain concepts

```
from pylab import *
t = linspace(0, .05, 300)

f = 50        # 50 Hz AC
Vm = 230 * sqrt(2)

y1 = Vm * sin(2*pi*f*t)
y2 = Vm * sin(2*pi*f*t + 120*pi/180)
y3 = Vm * sin(2*pi*f*t + 240*pi/180)

plot(t, y1)
plot(t, y2)
plot(t, y3)

plot(t, y2-y1, color='black')
show()
```

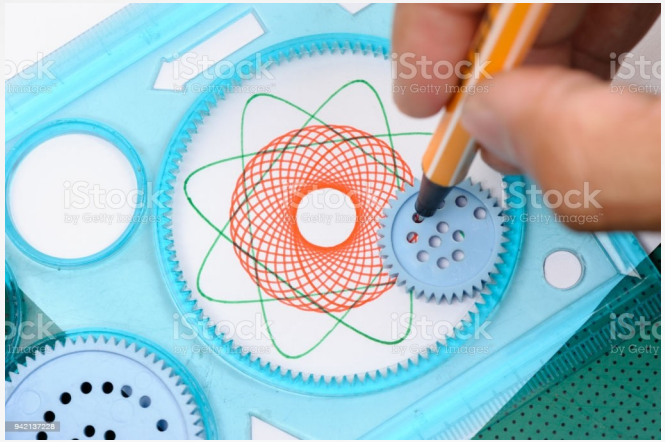


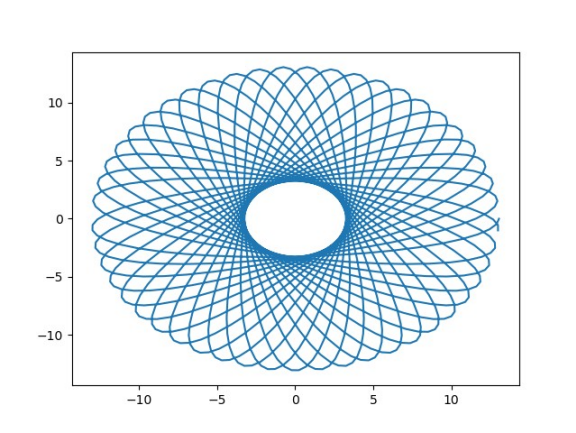
3 phase AC.

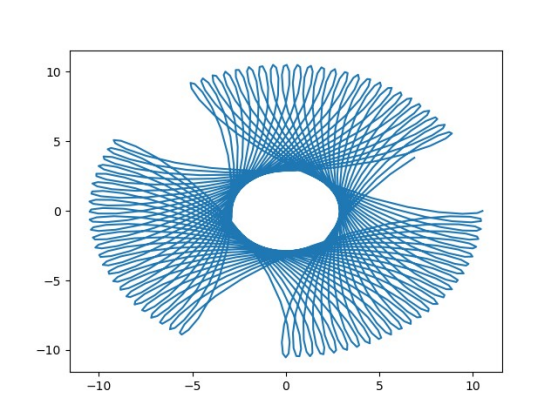
Nearly 400 volts between two phases.

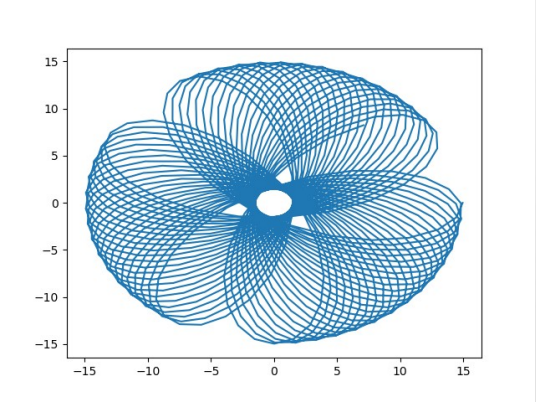
Spirograph (https://en.wikipedia.org/wiki/Spirograph)

```
from pylab import *

t = linspace(0, 50*pi, 1000)
r = 3.1
R = 10.

k = r/R
l = 1.6      # 1.5 and 2.5

x = R*((1-k)*cos(t) + l*k*cos((1-k)*t/k))
y = R*((1-k)*sin(t) - l*k*sin((1-k)*t/k))

plot(x,y)
show()
```



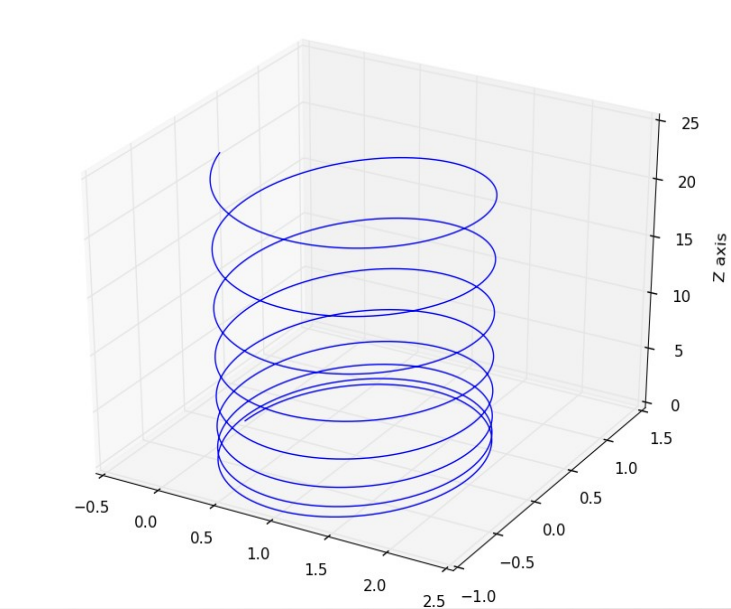






Simple simulations

```
from pylab import *
from scipy import integrate

m = 25.0       # Mass and
q = 5.0                  # Charge of the particle
E = array([0, 0, .1]) # Electric field components Ex,Ey & Ez
B = array([0, 0,  5]) # Magnetic field

def solver(X, t0): # X is a six element array, t0 required for the solver
        v = array([X[3], X[4], X[5]])  # make the velocity vector
        a = q * (E + cross(v,B)) / m   # F = q v x B ; a = F/m
        return [X[3], X[4], X[5], a[0], a[1], a[2]]

tmax = 50  # calculate up to 50 seconds
x0 = [0, 0, 0, 0, 1, 0]                          # position & velocity, at t = 0
t = arange(0, tmax, 0.01)                        # array of time coordinate
pv = integrate.odeint(solver, x0, t) # integrate for position and velocity

savetxt('xyz.txt',pv)          # saves 6 columns to file, x,y,x, Vx, Vy, Vz

from mpl_toolkits.mplot3d import Axes3D
ax = Axes3D(figure())
ax.plot(pv[:,0], pv[:,1], pv[:,2])               # 3d plot of x, y and z
ax.set_zlabel('Z axis')
show()
```



Trajectory of a charged particle

Advantages of Python

- Easy to learn
- Open Source
- Modules for
  - Scientific computation
  - Data Visualization
- Large user community
- Good Documentation

Resources Currently Available

Plenty of material on the web

- To learn Python programming
- To teach science using simulations

Not enough material to teach how to write small programs as a part of learning maths/science.

Some Possibilities….

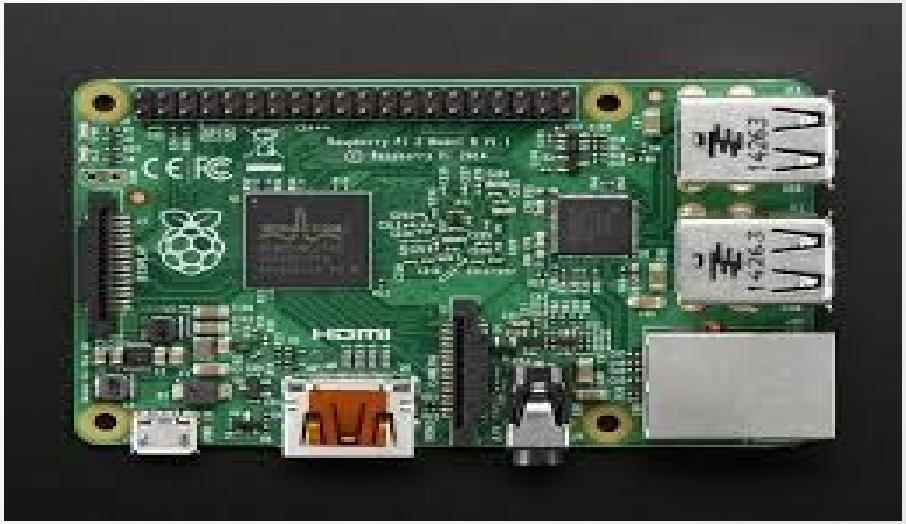
- Use Python to support the teaching of core subjects. Learning Python should happen as a byproduct.
- Make a collection of small (< 20 lines) Python programs related to the subjects covered in the school syllabus, along with documentation and videos.
- A Python companion for all text books at high school level.
- Documentation in Indian languages.

https://expeyes.in/python.html

School computers, save on hardware cost

For schools, do we really need 100 watts PCs with UPS etc., and tons of bloatware ? We need to teach:

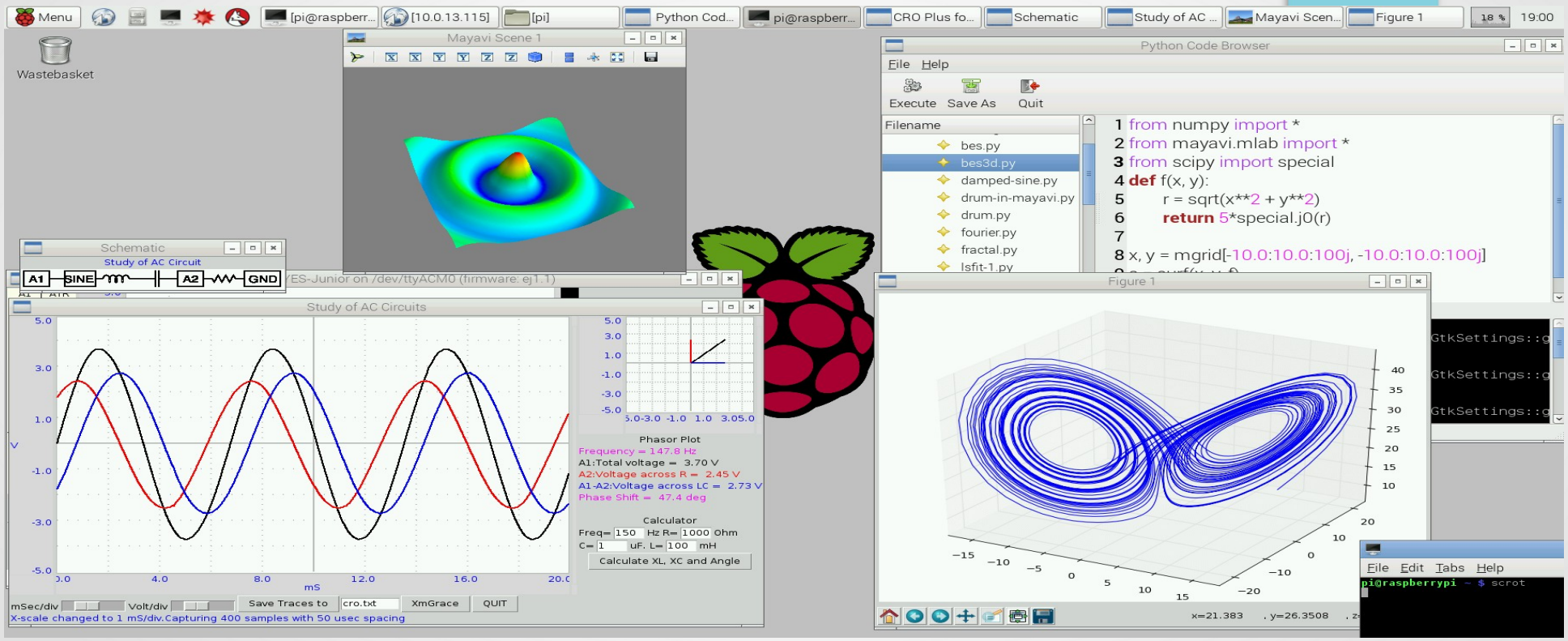
- Word Processing
- Programming languages
- Scientific computing
- Simulations
- Science Experiments
- Graphics ...etc.



Raspberry Pi 2

Costs around Rs. 10000/- , Consuming < 20 W , with monitor

ExpEYES, Matplotlib and Mayavi on Raspberry Pi 2



Other ARM boards, with 1GB RAM, like Cubieboard, Marsboard also works fine.

Arduino is very good, but it hides the micro-controller

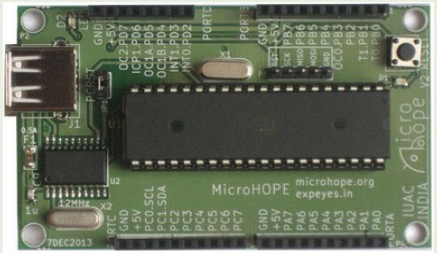
https://expeyes.in/microhope.html

ExpEYES … Your Lab@Home

Low Cost Science Experiments using Computers

Experiments | Design | How to Buy | Software | Android | python | MicroHOPE | People | News

Micro-controllers for Hobby Projects and Education (Latest Developement, KuttyPy)



MicroHOPE is a micro-controller development system based on Atmel ATmega32. Developed by IUAC, New Delhi, and released as open hardware. Unlike many other systems, the focus is on understanding the fundamentals of micro-controller architecture and programming rather than learning the features of the development kit. The knowledge gained on Atmega32 using MicroHOPE can be easily applied in understanding another micro-controller. There is no complex IDE for MicroHOPE. A simple text editor with a couple of extra buttons to invoke avrgcc and avrdude (to compile and upload the code) is all what we have. It is tested on both GNU/Linux and MS-Windows. Debian packages are available.

We start with simple C programs, treating the uC as a device with programmable Input/Output pins, and then introduce the peripherals like ADC, communication ports etc. The uC architecture is explored using simple assembly language programs and LEDs connected to the I/O ports.

User's manual, in PDF format, may be downloaded from here (or here)

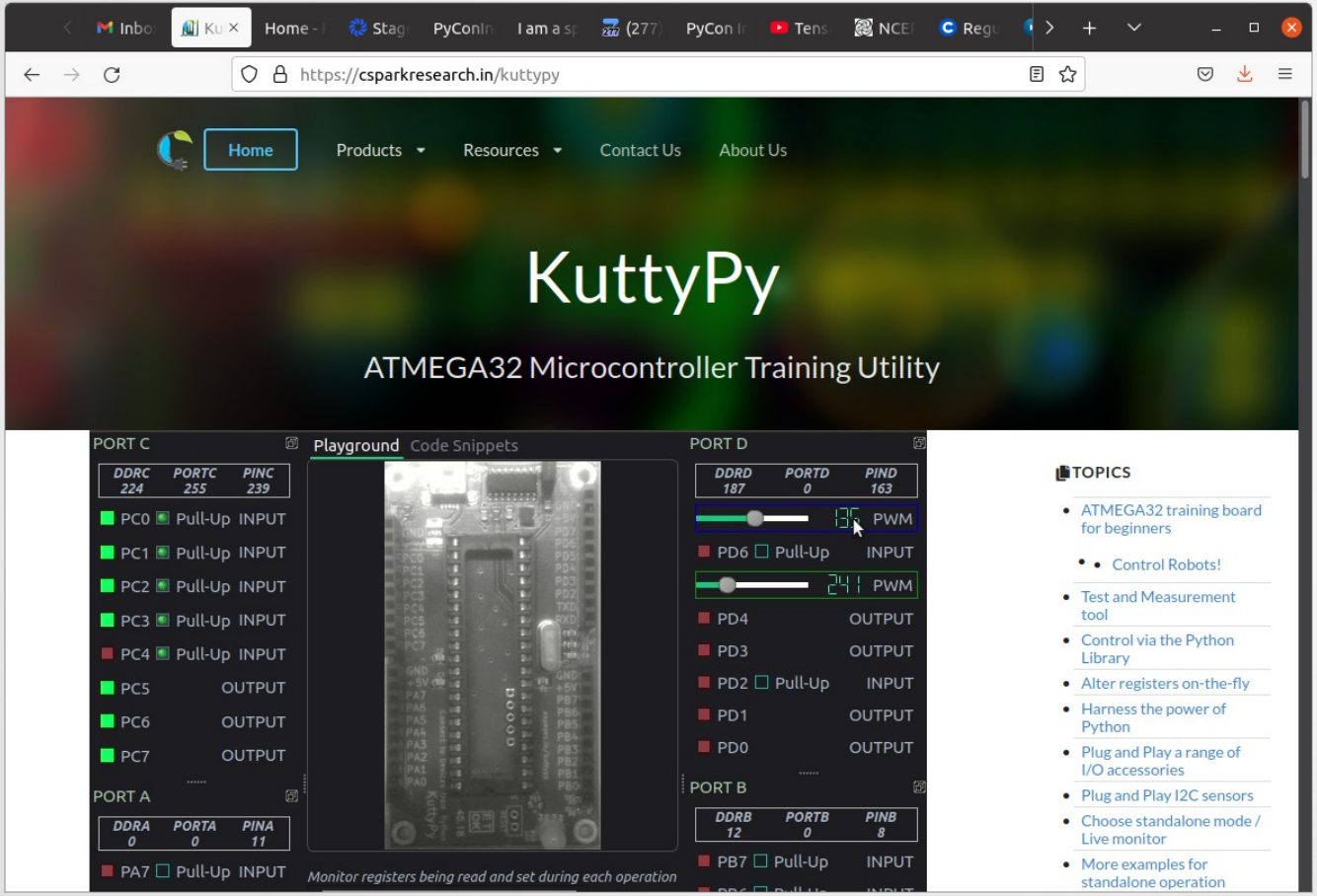
Hardware

Software

Library and Examples

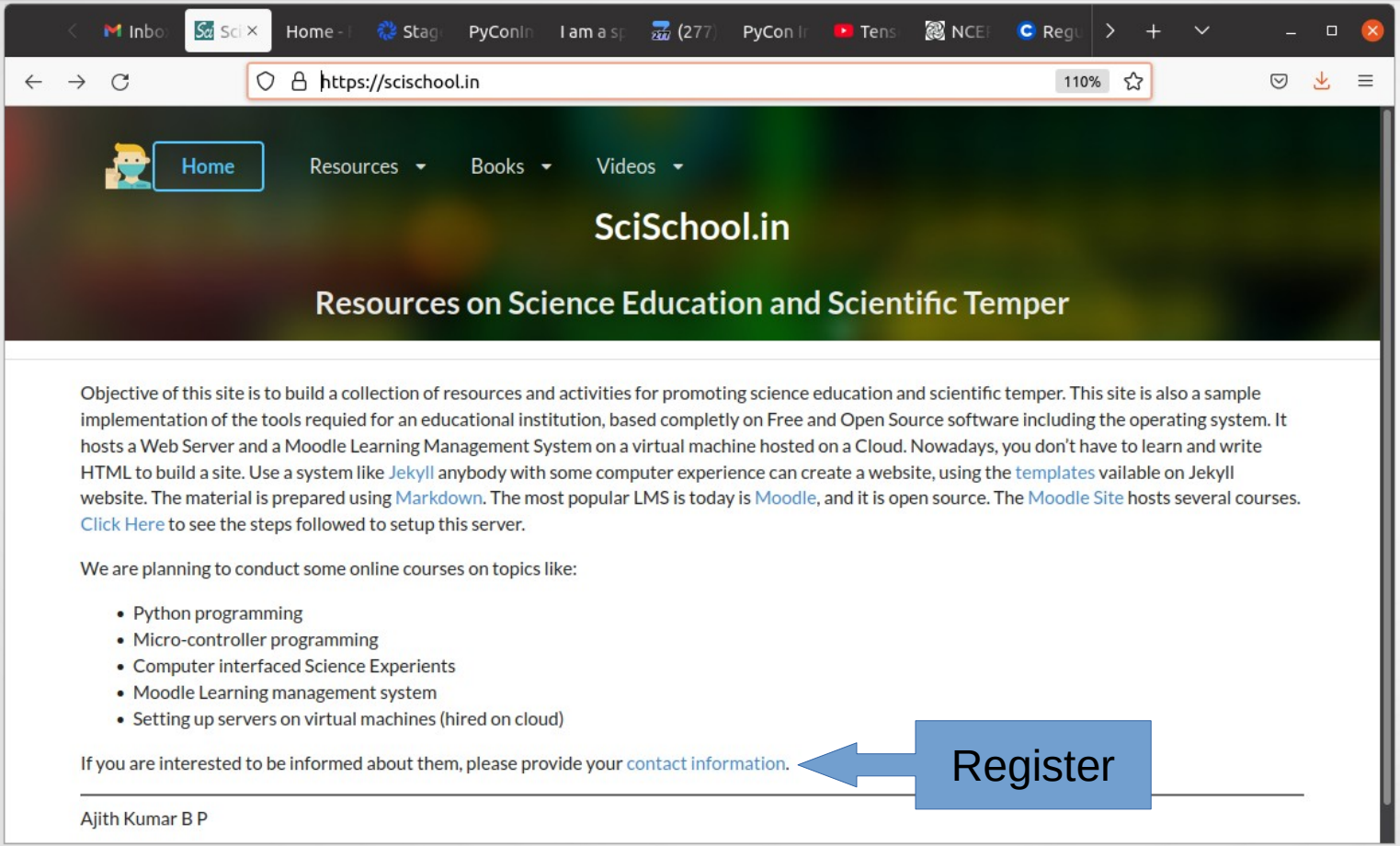
Assembler Examples

Python for Learning Micro-controllers (https://csparkresearch.in/kuttypy)



- AVR board connected to PC USB port
- Access all uC registers from Python
- Run Python Scripts
- Write code in C or Assembler
- Single Click Upload

https://scischool.in



Register

Sharing knowledge

- Python
- Micro-controllers
- Science Experiments
- Moodle LMS
- Servers on cloud
- Free/Open Software and Hardware

IUPAP: The International Union of Pure and Applied Physics

https://iupapc13.github.io

Home Resources

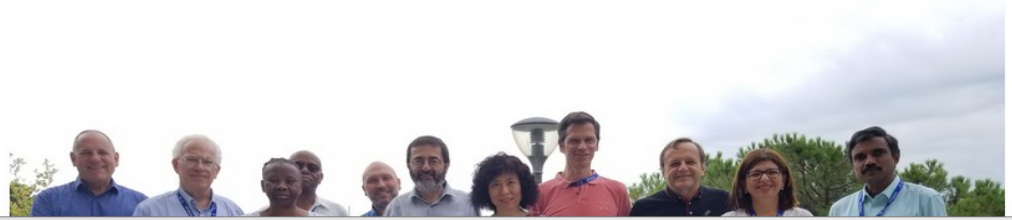
Affordable Scientific Instruments

Repository of IUPAP C13 (Physics for Development)

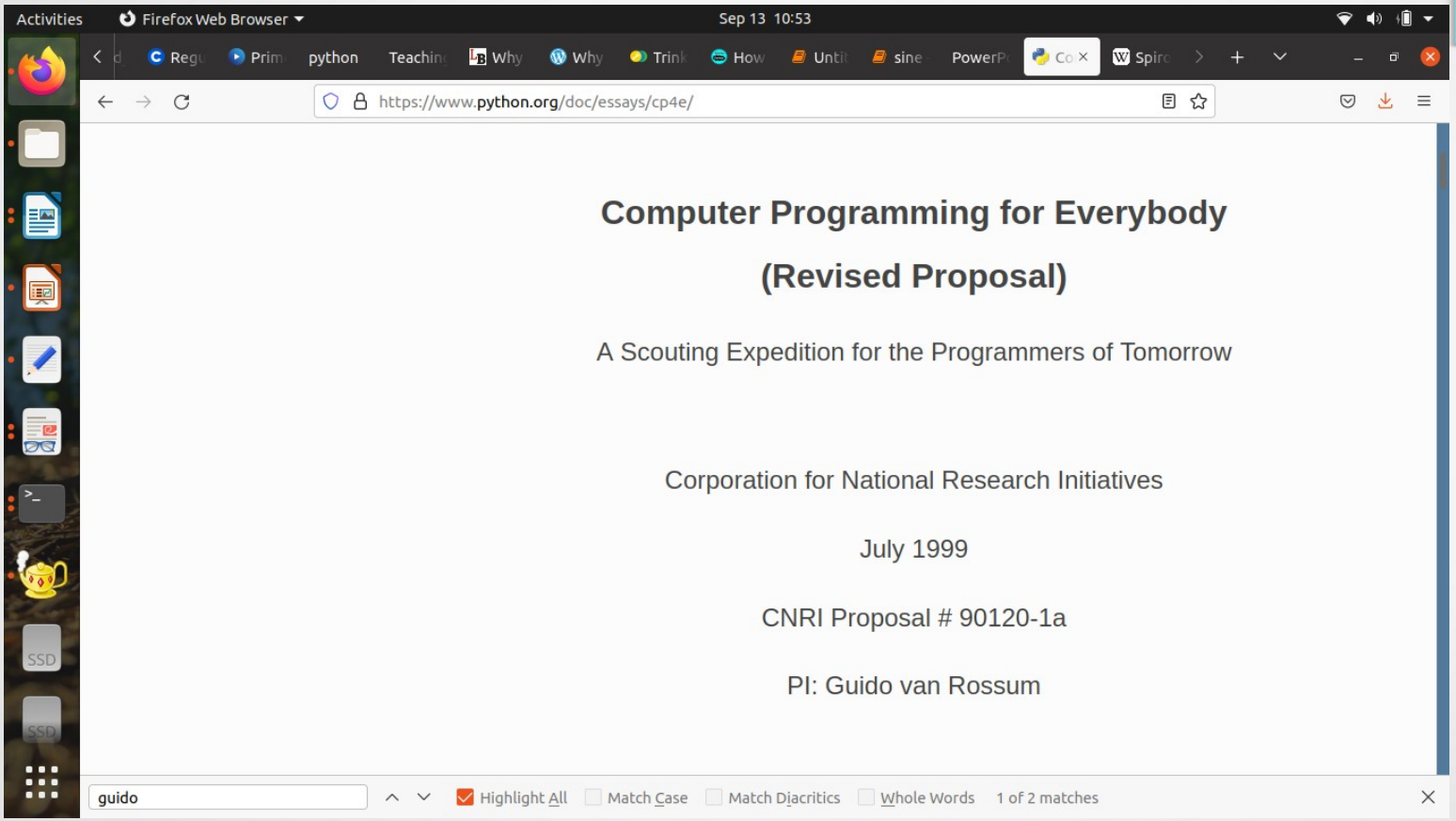
Experiment is what makes science different from other forms of knowledge, every hypothesis here must be verified by experiment. We need equipment to perform experiments. Lack of equipment is one of the reasons for the backwardness of experiments in science education. However, a large body of knowledge about the design and fabrication of scientific equipment exists in the public domain. There are open instruments applicable to various fields like education and health care. The IUPAP C13 commission attempts to explore this body of knowledge to:

- Create a thematic repository for Open Source projects in instrumentation and associated tools.
- Investigate the possibility of rewarding some excellent projects on the IUPAP web site.
- Launch collaborations between developing countries. Contests.
- Create a study group on sustainable-cost scientific instrumentation and laboratory hardware. This would be the first opportunity to lobby together for sustainable-cost instrumentation and laboratory hardware.
- Assemble the main experts in the field through dedicated Hackathons.

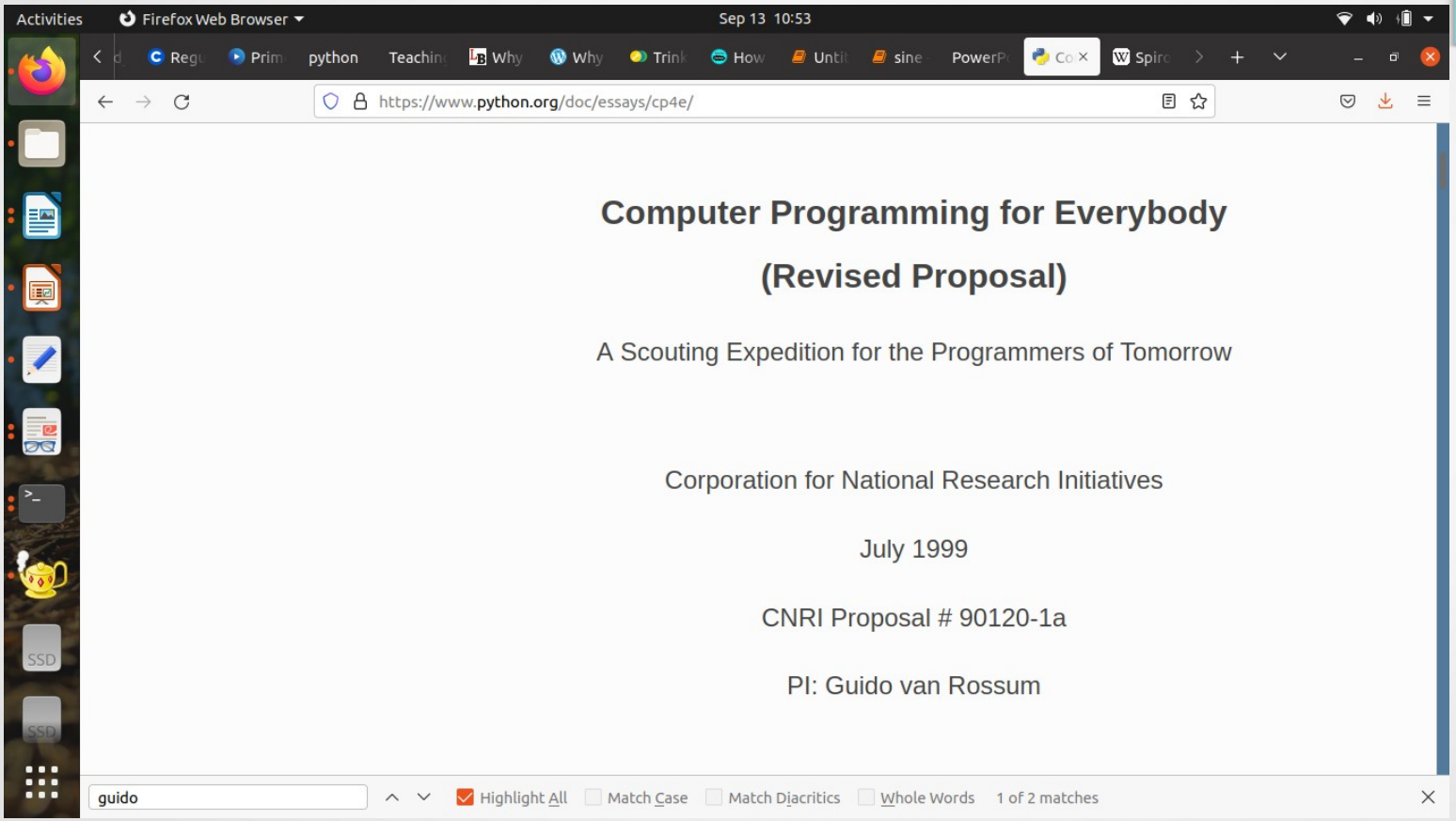
IUPAP C13 Commission Annual Meeting,Aug-2018, Trieste (Left-to-right: Joe Niemela (Sec'y), Michael Steinitz, Mmantsae Diale, Sekazi Mtingwa (Chair), Carlo Iorio, Fernando Quevedo (ICTP Director, Observer), Kuijuan Jin (Vice-Chair), Andreas Buchleitner, François Piuzzi, Samia Charfi Kaddour, Ajith Kumar B.P.



Enable Everyone...



Enable Everyone...



From CP4E proposal ...

We ask a follow-up question: "What will happen if users can program their own computer?" We're looking forward to a future where every computer user will be able to "open the hood" of their computer and make improvements to the applications inside. We believe that this will eventually change the nature of software and software development tools fundamentally.

We compare mass ability to read and write software with mass literacy, and predict equally pervasive changes to society. Hardware is now sufficiently fast and cheap to make mass computer education possible: the next big change will happen when most computer users have the knowledge and power to create and modify software.

From CP4E proposal ...

Our plan has three components:

- Develop a new computing curriculum suitable for high school and college students.
- Create better, easier to use tools for program development and analysis.
- Build a user community around all of the above, encouraging feedback and self-help.

These components come together in the scientific exploration of the role of programming in next generation computing environments.

We intend to start with Python, a language designed for rapid development. We believe that Python makes a great first language to learn: Unlike languages designed specifically for beginners, Python is also the choice of many programming professionals. It has an active, growing user community which has already expressed much interest in this proposal, and we expect that this will be a fertile first deployment ground for the teaching materials and tools we propose to create. During the course of the research we will evaluate Python and propose improvements or alternatives.

Programming, to teach all or not ....

Personal opinion:

“We do not have a policy to teach driving to only those who are planning to join Uber / Ola.

The same may be followed for programming”

Summary

- We need large number of good programmers.
- It requires exposing a larger number of people to coding, by some means.
- Teaching programming at school level to support the core subjects could be one option.
- Python is a language suitable for that purpose.
- The current system of teaching Office Packages in schools need to change (go).
- Python enthusiasts can help by preparing material to assist teaching of core subjects with the help of Python.
- Looking for collaborators.

Thank you