Lab. 10b

PicoBoard :: Port Calibration & Hall Effect

Step 1 – Calibrating a port

Step 2 – PicoBoard overview; Python API

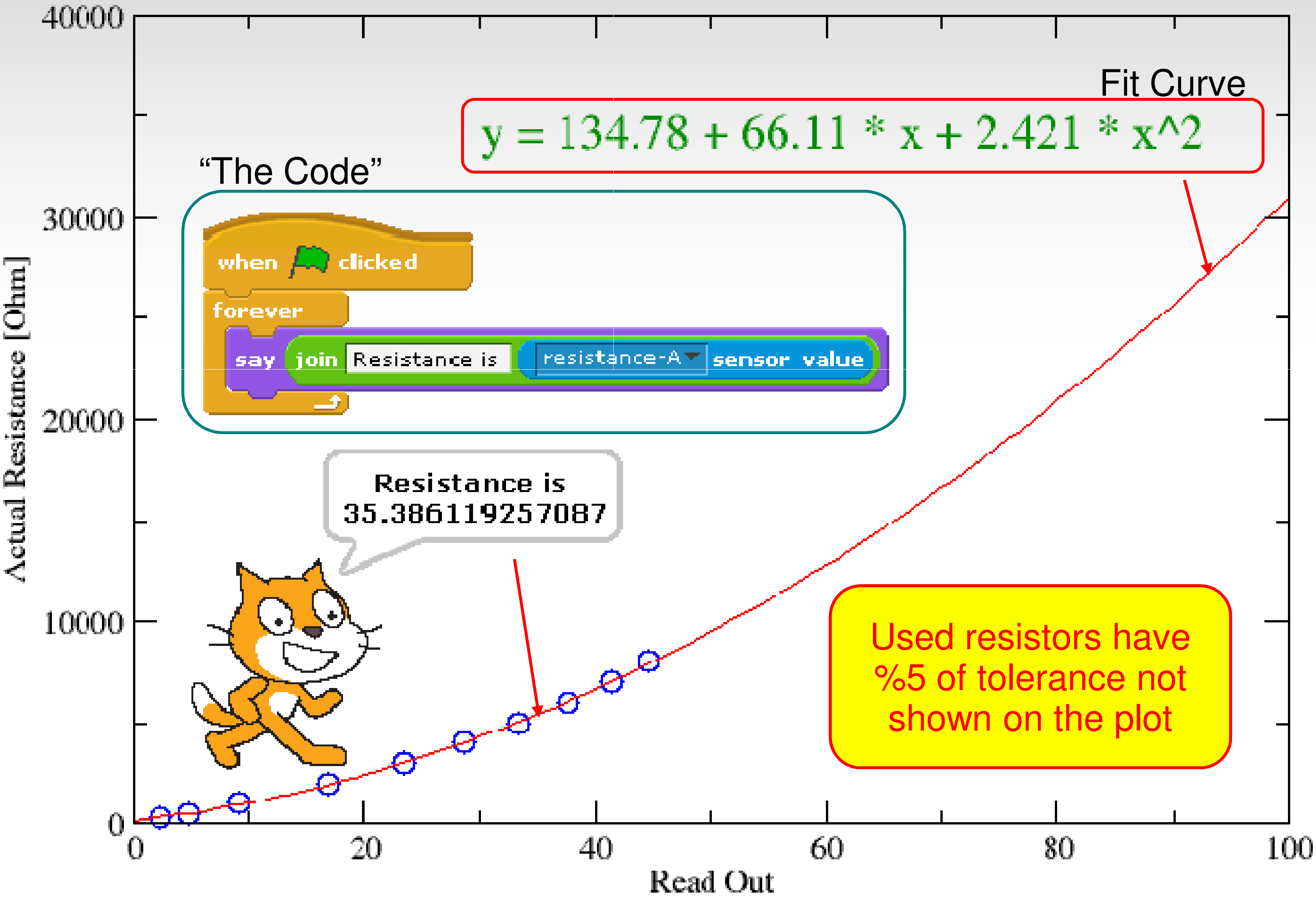
Step 3 – Hall effect measurement; distance calibration

International School on Trigger and Data Acquisition, 1-7 February 2010, Ankara, Türkiye

PicoBoard 10b – Dan Savu & Özgür Çobanoğlu

Calibrating a Port

Resistance reading port A (B, C, and D)



Fit Curve

$$y = 134.78 + 66.11 * x + 2.421 * x^2$$

“The Code”

```
when 🟩 clicked
forever
    say join Resistance is resistance-A ▾ sensor value
```

```
Resistance is
35.386119257087
```

Used resistors have %5 of tolerance not shown on the plot

Calibrating a Port

Resistance reading port A (B, C, and D)

To Do:

- Change directory to "***Lab10b/Step_1***"
- Start the application: "Scratch"
- Create the simple code of previous page
  - ***Drag & drop*** items from left pane
  - Click on the graphical script or the ***green flag*** to run it
- Use known 3 kOhm resistors to calibrate one of the ports
  - Connect known resistors and record what are read-out to form a ***two column data file*** (e.g. file.dat) where the first column is the value read-out from the port and the second one is the known resistor value
  - Plot this data with the Curve Expert application
  - ***Fit*** a ***second order curve*** (i.e. calibration curve) to the measured data points (e.g. ***Tools*** **>** ***Curve Finder*** **>** ***All On/ OK***)
  - Extract/save the ***parameters*** of the fit curve
- Connect the unknown resistor to the port and measure its resistance according to the calibration curve

International School on Trigger and Data Acquisition, 1-7 February 2010, Ankara, Türkiye
PicoBoard 10b – Dan Savu & Özgür Çobanoğlu

Lab. 10b

PicoBoard :: Port Calibration & Hall Effect

Step 1 - Calibrating a port

Step 2 – PicoBoard overview; Python API

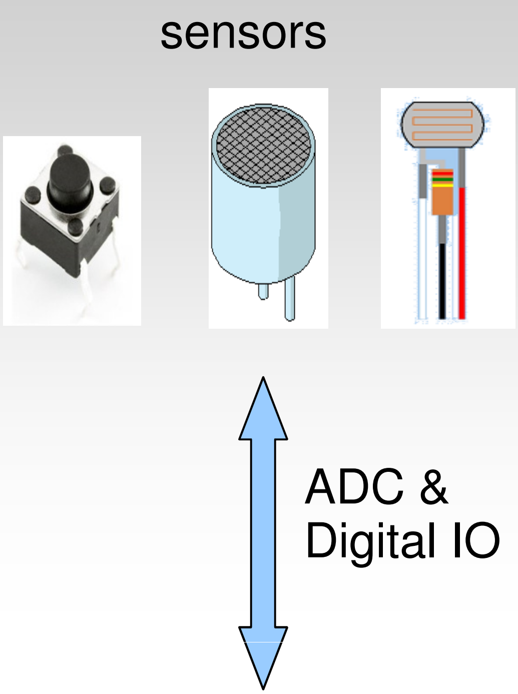
Step 3 – Hall effect measurement; distance calibration

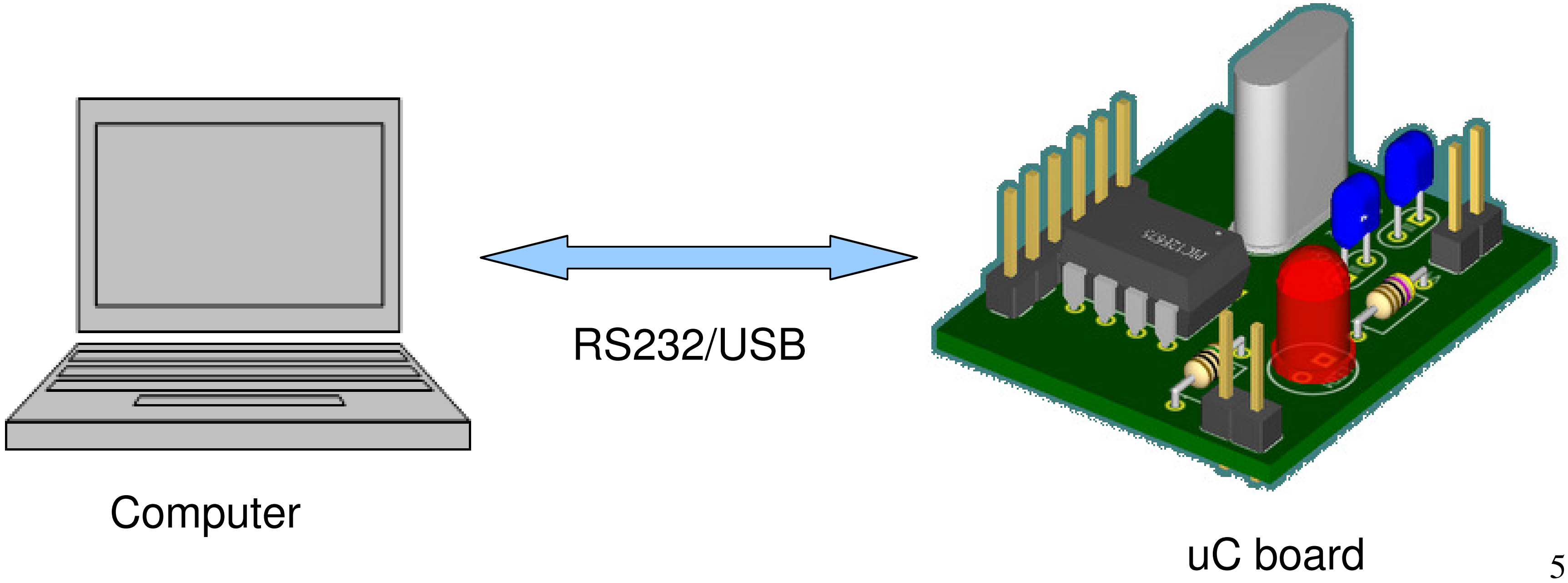
International School on Trigger and Data Acquisition, 1-7 February 2010, Ankara, Türkiye
PicoBoard 10b – Dan Savu & Özgür Çobanoğlu

PicoBoard overview

hardware at a glance

- PIC 16F676 microcontroller
  - 8bit CPU @ 20MHz
  - 5V, 100uApower features
  - 10bit resolution ADC
- RS232 communication protocol
- A set of sensors
- 12 IO channels

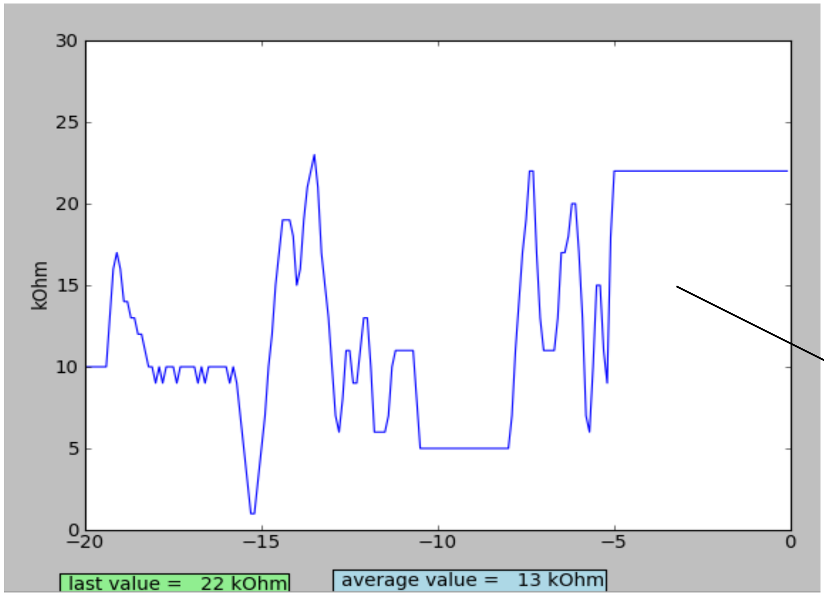




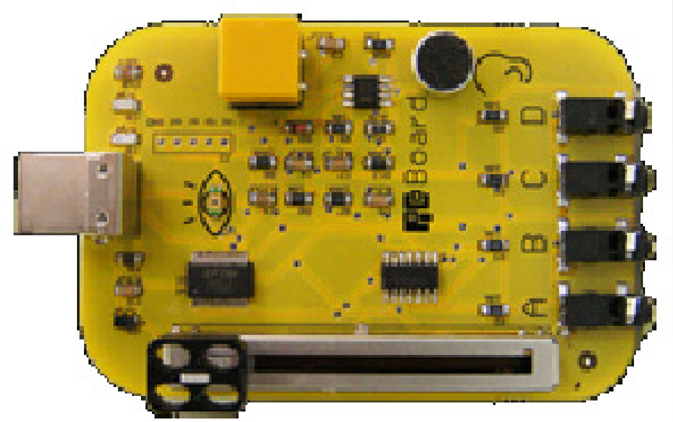
PicoBoard overview

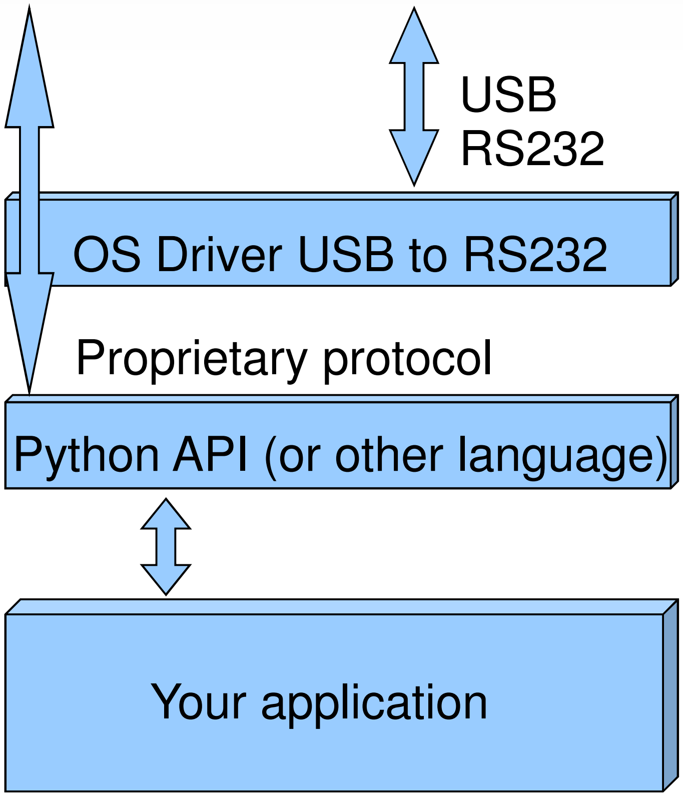
software at a glance

- Standard serial interface
- Simple Proprietary protocol
- Python API
  - ScratchBoard(serialPortNo)
  - sb.open()
  - sb.close()
  - sb.getSensorValues()
  - sb.getSliderValue()
  - sb.getButtonValue()
  - sb.getLightValue()
  - sb.getSoundValue()
  - sb.getResistanceXValue() ; X=A, B, C, D



PicoBoard





USB
RS232

OS Driver USB to RS232

Proprietary protocol

Python API (or other language)

Your application

International School on Trigger and Data Acquisition, 1-7 February 2010, Ankara, Türkiye
PicoBoard 10b – Dan Savu & Özgür Çobanoğlu

PicoBoard API

Resistance calibration

To Do:

- Change directory to "***Lab10b/10b_resistance_calibration***";
- Open A_pico.py and A_resistance_calibration.py using notepad++;
- Have a look at the python code in those 2 files;
- Adjust the getResistanceAValue(…) function in A_pico.py with the calibration formula you've just computed. (line 81);
- Adjust the SERIAL_PORT variable value with the com port number used for PicoBoard communication (line 8, file A_resistance_calibration.py)
- Open a new command console;
  (notepad++: ***Run > Open current dir cmd***)
- Run the A_resistance_calibration python program;
  (console: ***python A_resistance_calibration***)
- Have a look at the graph and measure again both known and unknown resistances.

International School on Trigger and Data Acquisition, 1-7 February 2010, Ankara, Türkiye

PicoBoard 10b – Dan Savu & Özgür Çobanoğlu

Lab. 10b

PicoBoard :: Port Calibration & Hall Effect

Step 1 - Calibrating a port

Step 2 – PicoBoard overview; Python API

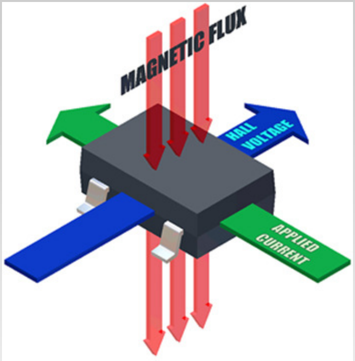
Step 3 – Hall effect; distance measurement calibration

Hall effect

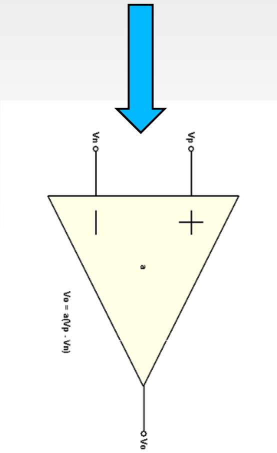
Measurement using Picoboard

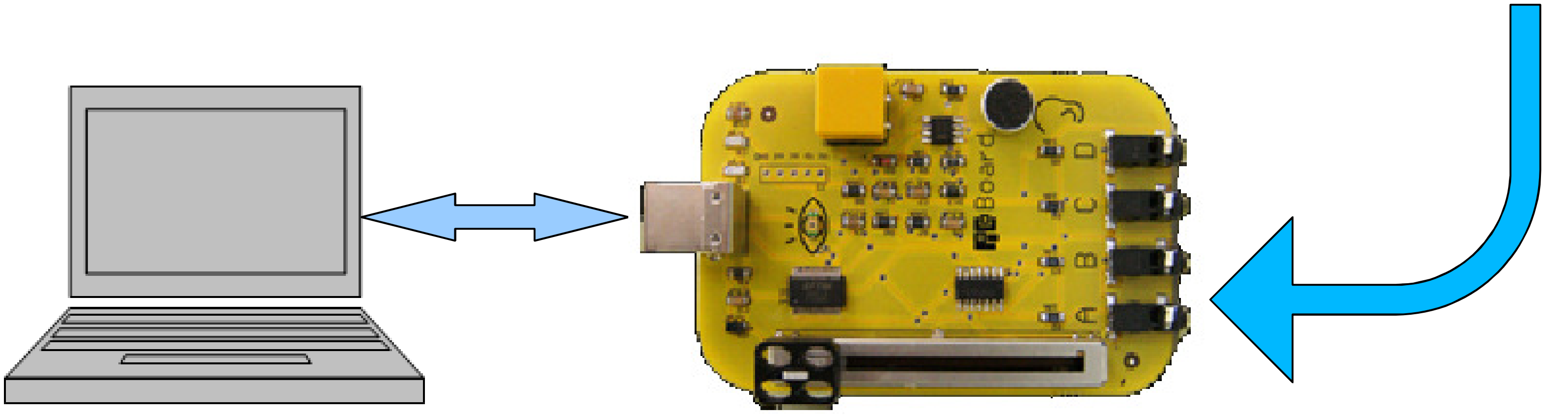
- The magnetic field intensity is converted into voltage by the hall effect sensor;
- Because the hall effect sensor voltage is too low we have to use an operational amplifier to amplify it;
- Next the amplified voltage will be converted into a digital value by the picoboard ADC;
- The voltage value will be transferred to a PC via the USB/RS232 interface.

Hall Effect Sensor



Voltage Amplifier





Computer

PicoBoard

International School on Trigger and Data Acquisition, 1-7 February 2010, Ankara, Türkiye
PicoBoard 10b – Dan Savu & Özgür Çobanoğlu

Hall effect

Port calibration (distance measurement)

To Do (part 1):

- Start the current supply and set the value to 5mA;
- Connect the hall probe picoboard to one of the computer USB ports;
- Change directory to "***Lab10b/10b_hall_probe***";
- Open B_pico.py and B_hall_probe.py using notepad++;
- Have a look at the python code in those 2 files;
- Adjust the getResistanceAValue(…) function in A_pico.py with the calibration formula you've just computed. (line 81);
- Adjust line 54 of B_hall_probe.py file so that the if statement is true whenever someone press the picoboard button;
- Open a new command console;
  (notepad++: ***Run*** **>** ***Open current dir cmd***)

Hall effect

Port calibration (distance measurement)

To Do (part 2):

- Run the A_resistance_calibration python program;
  (console: ***python A_resistance_calibration***)
- Have a look at the graph while trying to approach the magnet to the hall effect sensor;
- Measure how the voltage read by the picoboard varies with the magnet distance and write down the values (voltage and distance, 10 samples);
- Try to find out a curve (equation) that acomodates the data you've collected. For this you can use the CurveExpert application on the desktop;
  (e.g. ***Tools* > *Curve Finder* > *All On/ OK***)
- Define a new function in B_pico.py called getDistance() which will return the distance between the hall effect sensor and the magnet in mm;
- Adapt B_hall_probe.py to use this new function to plot the new value;

Homework

- Try to find out some other applications for the picoboard;
- Write down a detailed explanation of your design + schematics;
- Eventually implement your idea and make some pictures (not mandatory) ;