Sensory Robotics - IR distance measurement

Goals:

- distance measurement on the basis of IR light-reflection;
- to become familiar with the signal-distance characteristics of IR photodiode and IR phototransistor.

Short description of the exercise:

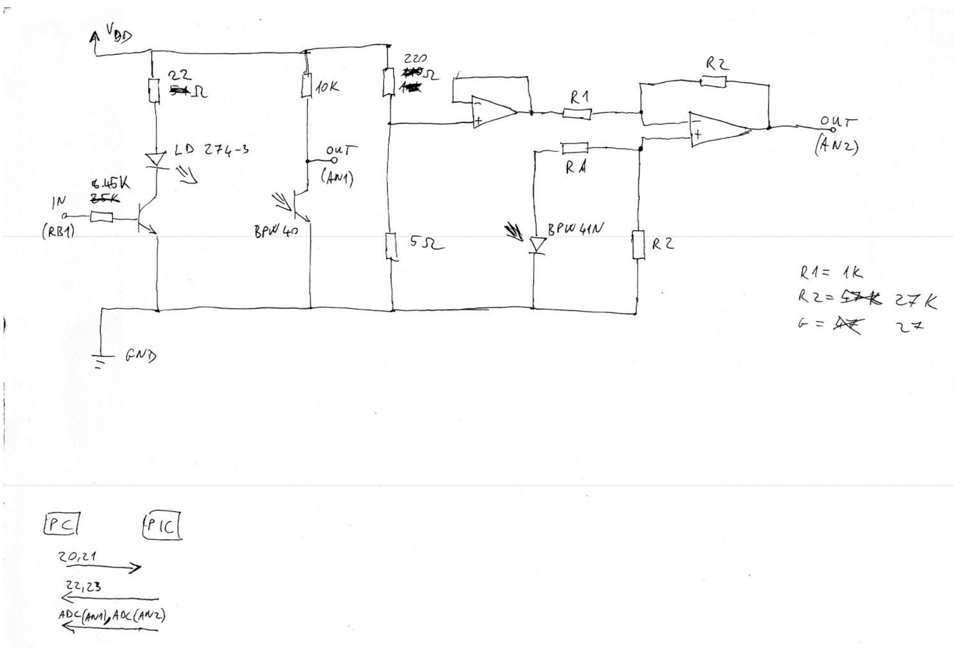
Software environment: Python.
Tools used during the measurement:

- already prepared circuit on a breadboard;
- xyz-table (interfaced to Python, through serial line / RS-232);
- power supply.

The xyz-table will bring a flat surface closer and closer to the circuit, from a perpendicular direction. The circuit is capable to measure the distance between the flat surface and the circuit itself, on the basis of the intensity of light reflection.
We should forward the raw, analog-digital converted (ADC) value of both the photodiode and the phototransistor to the personal computer, in order to prepare a distance versus signal level characteristics. At the end of this lab, the main goal is the prepared figure containing this distance - signal level curve (of course, together with your source code and your report).

Description of the circuit:

The following circuit can be found on the breadboard (without the controlling part):



The microcontroller (PIC 18f2321) controls the transistor through its RB1 pin, which transistor switches on the light emitting diode LD 274-3. The reflected light of this LED can be measured by the phototransistor BPW 40 and the photodiode BPW 41N:

- The phototransistor (BPW 40) is similar to a bipolar transistor, but its base is built up as a light-sensitive surface, so the opening-current (charge-flow) is induced by the incoming light. The resistor (10 KOhm) is in series combination with the phototransistor, so in the point between them (OUT-AN1) we make it possible to measure the through-flowing current on the phototransistor as a voltage-level.
- The output of the photodiode (BPW 41N) can produce few-hundreds of milli Volts, so we need a difference amplifier to make this signal measurable to the microcontroller: the first OpAmp (called unity follower) stabilizes the voltage level produced by the resistors 220 Ohm and 5 Ohm; the second OpAmp realizes the difference amplifier itself. The output at OUT-AN2 will be a voltage level, which can be directly measured by the inner ADC of the microcontroller.

Theoretical assumptions when computing circuits containing operational amplifiers:

- there is no in-coming current on the +/- input pins of the OpAmps, only voltage levels are present,
- if we have a closed-loop circuit (the output is connected to one of the input terminals), then the OpAmp will do its best to hold its two input terminals on equipotential levels (as far as its supply power makes it possible, of course).

How does the differential amplifier work:
https://en.wikipedia.org/wiki/Operational_amplifier_applications#Differential_amplifier_.28difference_amplifier.29

How does the ADC of the microcontroller work (in our case):
the analog input voltage level in the range 0V - 5V is quantized to 10 bits (1024 values), from which our measurement environment sends only the 8 upper bits (so every measurement data can have 256 different values, as a number represented on 8 bits).

Description of the measurement:

During this lab, we will control everything from Python, which will communicate with the board on serial line (RS-232). The microcontroller on the board will switch on the LED automatically, will measure the output signal level of the photodiode and the phototransistor, then it will send back the two values to the computer, according to the following protocol:
request for measurement: PC ---> microcontroller: ‘20’, ‘21’
result of measurement: microcontroller ---> PC: ‘22’, ‘23’, {ADC of AN1}, {ADC of AN2}

Detailed steps to do:

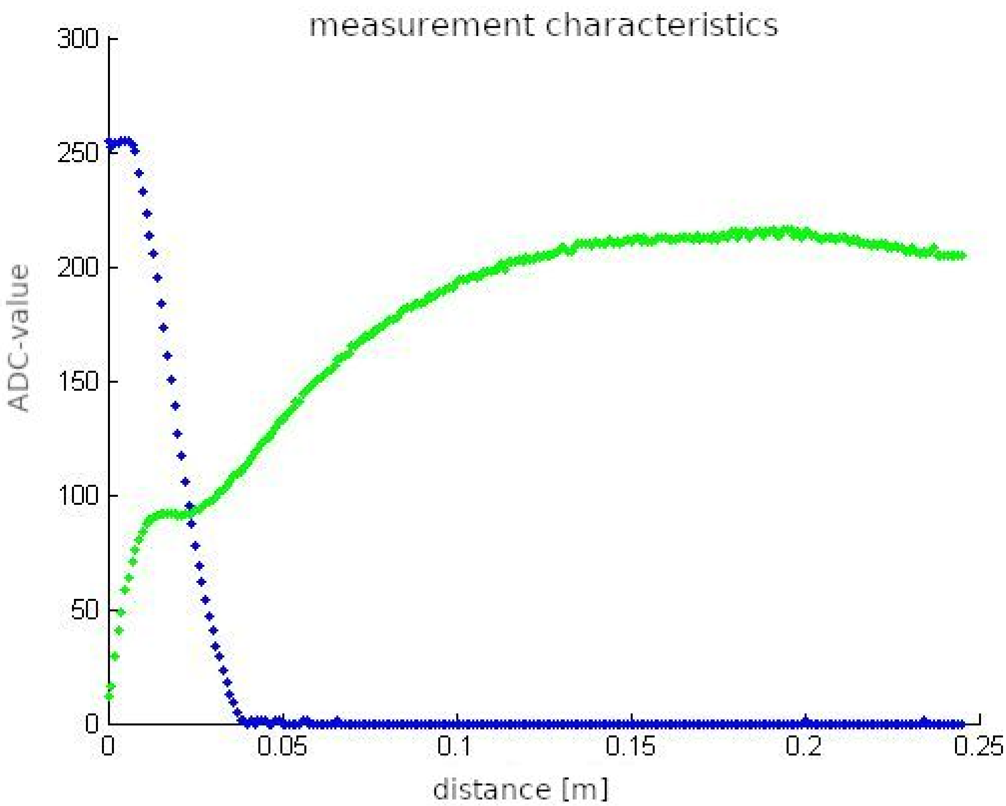
1. Please read and interpret the file **`sp_c.py`**: this class is responsible for the communication with the microcontroller. In its **`measure`** function it realizes a single iteration of the communication. *There is no need for coding at this step.*
2. Please overview the file **`Marci_PyPowerCube.py`**: this class is responsible for the low-level communication with the xyz-table (original source:

https://github.com/naszy/PyPowerCube). (This is an excellent example of how to realize a low-level interface to a physical machine through serial communication.) *There is no need for coding at this step.*

3. Please open the given framework-file: **`infra_test.py`**
   a. this file takes care of the xyz-table (on high level):
      i. initialization,
      ii. moving the end-cube to the start-position,
      iii. iterative instruating of the measurement itself,
      iv. closing the device;
   b. please edit the 3rd block of this framework-file, iteratively sinking module z, and at every depth-level please measure with our previously written communication-function.

During online education, you do not have to complete this file, the results of the measurement is given to you as **`*.npy`** files. Just please understand what happens here from step to step.

4. Please extend the script **`distance_measurement_plot.py`** which produces a figure from the saved measurement data, similarly as below:



Available source-codes:

Already given:

- **`sp_c.py`**: serial communication with the microcontroller;
- **`Marci_PyPowerCube.py`**: serial communication with the xyz-table;

- **`infra_test.py`**: the framework-file to the complete measurement, normally you have to complete it, but during online education just interpret it;
- **`numpy_array_of_infra_measurement_alu.npy`**: recorded measurement data, with a shiny aluminium-foil;
- **`numpy_array_of_infra_measurement_paper.npy`**: recorded measurement data with a white paper.

Please implement / extend:

- **`distance_measurement_plot.py`**: a script, which draws a figure about the characteristics of the sensors, on the basis of the recorded measurement-file.

What and when to upload to moodle:

What:

- the plotter file (distance_measurement_plot.py),
- your report.

Please include in the report the calculations regarding the differential amplifier around the photodiode (BPW 41N):

- what potential level is set up between the two resistors, at the input terminal of the unity follower,
- what is the value of the gain of the second OpAmp,
- what is the range of potential levels appearing at the photodiode, when the OUT-AN2 terminal is in the linear (not saturated) region between [0, 255] (approximately the distance range between 1-4 cm in the characteristics above).

Please also make some comments about the characteristics of both sensors, considering the datasheets in the working directory.

Deadline: indicated in moodle.

Thank you.
Miklós