University of Montenegro
Faculty of Electrical Engineering

Course: Industrial electronics
Theme: Vernier hand dynamometer

Mentor :
Prof. dr Radovan Stojanović

Students:
Milojka Ratković 01/19
Marko Katić 02/19
Mihailo Kopitović 37/19

Date and place: April 2020, Podgorica

Contents

Problem description

The task of this project is to demonstrate sampling process of a Vernier dynamometer sensor at time interrupt levels with the possibility of setting the START / STOP function and sampling frequency at 5Hz, 2Hz and 1Hz. In our case, this set of commands defines the sampling rate of the sensor. This sampling can be taken every 0.2sec, 0.5sec and 1sec.

Connection scheme

Our hand dynamometer can be used to measure grip strength, pinch strength, and to perform muscle exhaust studies (that is why this sensor is often used in medicine). The hand dynamometer sensor can be used with other sensors (e.g. EKG sensor) to study muscular health and activity. Displayed measured values of force are performed with Arduino Uno.

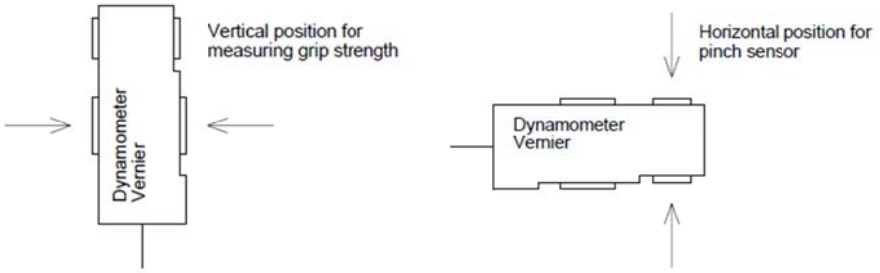


Figure 1.0 – Measuring grip and pinch strength

Grip strength is assessed holding the sensor in a vertical position with the arm perpendicular to the body and fingers on the pad distal to the longer proximal portion to which the palm is pressed. Pinch strength is assessed by placing the sensor on a flat horizontal surface with the pinch sensor pads in a vertical alignment, extending beyond the table plane.

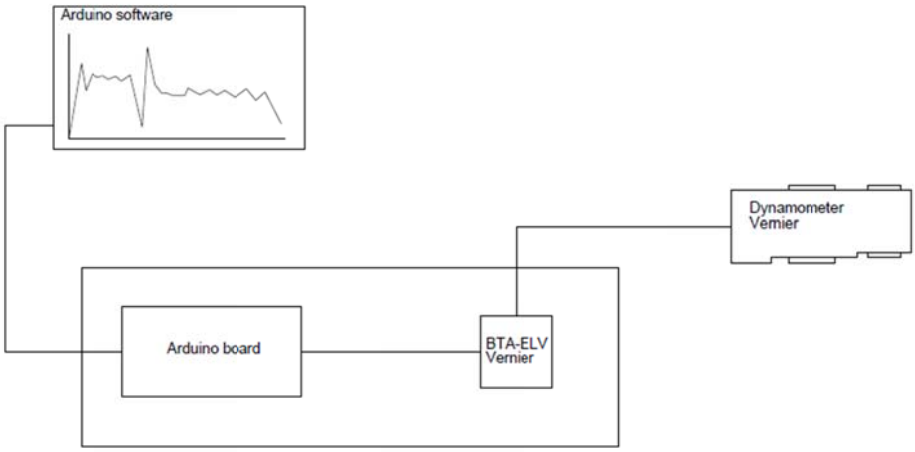


Figure 1.1 – Connection scheme

The following Vernier dynamometer is connected to an Arduino Uno board through Vernier BAT-ELV (analog protoboard adapter), while samples on Serial Monitor and their graphical representation are shown in the Serial Plotter of Arduino software.

Program

Arduino timer interrupts allows us to briefly pause the normal sequence of events taking place in the loop() function at precisely timed intervals, while we execute a separate set of commands. Once these commands are done the Arduino picks up again where it was in the loop().

Arduino code:

```
#include <EEPROM.h> //  this is needed to access EEPROM memory

float voltage = 0;
double Force = 0;

char inChar = 'A';
int sensorValue = 0;
unsigned int Nt = 53036;
boolean read = false;
unsigned int NtEeprom=0;

void setup() {
  Serial.begin(57600);
  NtEeprom = ((((EEPROM.read(4))*10 + EEPROM.read(3))*10 +
EEPROM.read(2))*10 + EEPROM.read(1))*10 + EEPROM.read(0); //returns the
number on that adress
  Nt = NtEeprom;
  noInterrupts();           // disable all interrupts
  TCCR1A = 0;
  TCCR1B = 0;

  TCNT1 = Nt;            // preload timer 65536-16MHz/256/XHz / X Hz is
sampling frequency
  TCCR1B |= (1 << CS12);    // 256 prescaler 64us
  TIMSK1 |= (1 << TOIE1);   // enable timer overflow interrupt
  interrupts();             // enable all interrupts
}

void loop() { //begining of loop
   if(read && inChar == 'a') //If A/D read – press ‘a’ for start
  {
    Serial.println(Force); //Print value
    read = false; //enable A/D read again
  }

  if(read && inChar == 'b') //If A/D read –press ‘b’ for sampling freq 5Hz
  {
    Nt = 53036; // T=0.2s
    EEPROM.write(0, 6); // write 53036 in EEPROM
    EEPROM.write(1, 3);
    EEPROM.write(2, 0);
    EEPROM.write(3, 3);
    EEPROM.write(4, 5);
    Serial.println(Force); //Print value
    read=false; //enable A/D read again
```

```
  }
  if(read && inChar == 'c') //If A/D read –press ‘c’ for sampling freq 2Hz
  {
    Nt = 34286; // T=0.5s
    EEPROM.write(0, 6); // write 34286 in EEPROM
    EEPROM.write(1, 8);
    EEPROM.write(2, 2);
    EEPROM.write(3, 4);
    EEPROM.write(4, 3);
    Serial.println(Force); //Print value
    read = false; //enable A/D read again
  }
  if(read && inChar == 'd') //If A/D read –press ‘c’ for sampling freq 1Hz
  {
    Nt = 3036; // T = 1s
    EEPROM.write(0, 6); // 3036 in EEPROM
    EEPROM.write(1, 3);
    EEPROM.write(2, 0);
    EEPROM.write(3, 3);
    EEPROM.write(4, 0);
    Serial.println(Force); //Print value
    read = false; //enable A/D read again
  }
  if(read = false && inChar == 'e') //If A/D read –press ‘e’ to stop
  {
    Serial.println(Force); //Print value
    read = true; //enable A/D read again
  }
}

void serialEvent(){
  while(Serial.available()){
    inChar = (char)Serial.read();
    }
  }

  ISR(TIMER1_OVF_vect){
  TCNT1 = Nt;            // preload timer
  sensorValue = analogRead(A0); //read A/D
  voltage = float(sensorValue)*5/1024;
  Force=-1.6641+17.8875*voltage; // Shows kilograms
 read=true;                 //A/D read
}
```

For sampling frequency of f_o=5Hz every 0.2 seconds a new sample is printed on the Serial Monitor, for f_o=2Hz every 0.5 seconds, and for f_o=1Hz every 1 second. A more simple preview of this idea is shown in the table below.

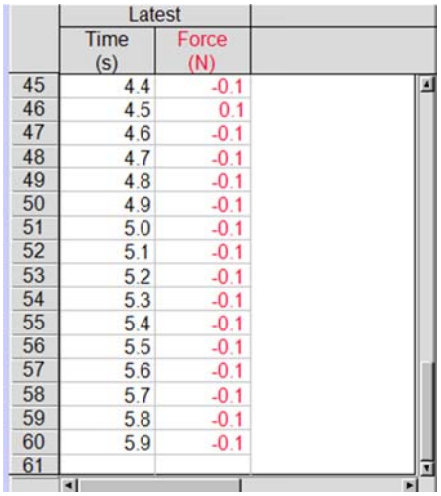
Input	Function/Sampling frequency	Function/Sampling period
a	START	START
b	f_o=5Hz	T=0,2sec
c	f_o=2Hz	T=0,5sec
d	f_o=1Hz	T=1sec
e	STOP	STOP

Table 1.0 – Input characters on Serial Monitor and achieved sampling

Diagram

As proof that our code solution is vaild, compare sample values we get from software Logger Lite 1.9.4 and Arduino to demonstrate on an simple example that our sensor is precise enough.
The maximum allowed variation in accuracy with the Vernier hand dynamometer is ±0.6N.

As we can see in the following two graphs (figures 2.2, 2.3), the sensor does not need to be calibrated in Logger Lite, so we will use it to verify that our dynamometer function and code are valid.
At steady state, Logger Lite shows through time a small shift of -0.1 instead of 0.00, while the Arduino mostly shows -0.0, and only sometimes 0.08. We conclude that the calibration has been successfully completed and that our solution to the problem is acceptable.



	Latest	
	Time (s)	Force (N)
45	4.4	-0.1
46	4.5	0.1
47	4.6	-0.1
48	4.7	-0.1
49	4.8	-0.1
50	4.9	-0.1
51	5.0	-0.1
52	5.1	-0.1
53	5.2	-0.1
54	5.3	-0.1
55	5.4	-0.1
56	5.5	-0.1
57	5.6	-0.1
58	5.7	-0.1
59	5.8	-0.1
60	5.9	-0.1
61		

Figure 2.2 – Measuring value in steady state on Logger Lite 1.9.4 software

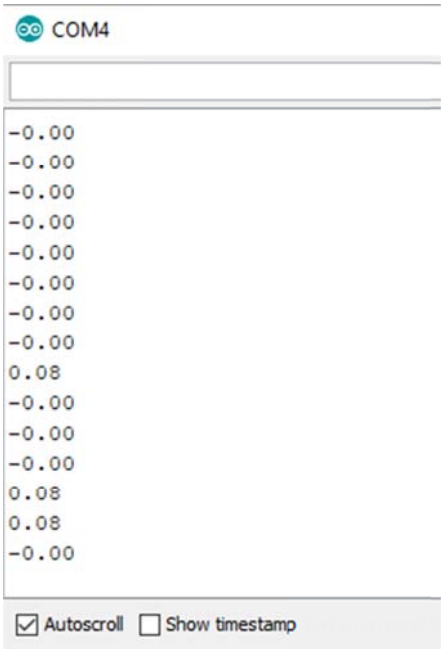


Figure 2.3 – Measuring value in steady state on Arduino software

After the validating is done, we can test our pinch/grip strength.

As a software solution we used Arduino. For showing the frequency sampling, Serial Monitor is used (path: Tools/Serial Monitor).
This paper will not show samples from Serial Monitor because it requires a video demonstration, but it can be simply explained: when the character "a" is entered in the command window, after pressing Send, the sequence starts, and we have values for different pressure applied. The speed of printed values depends on the sampling frequency.

Strength can be measured by applying pressure for a series of short grasps or over a sustained duration. On figures 2.4, 2.5 Serial Plotter on Arduino Uno is shown for sampling frequency of 5Hz. We can notice that the mesaurement is sufficiently precise because it shows the little variations of the signal even for a approximately sustained duration.

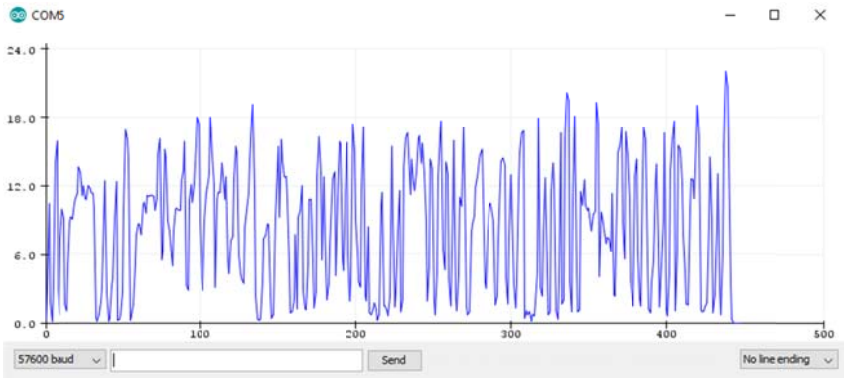


Figure 2.3 – Arduino Serial Plotter results for series of short grasps

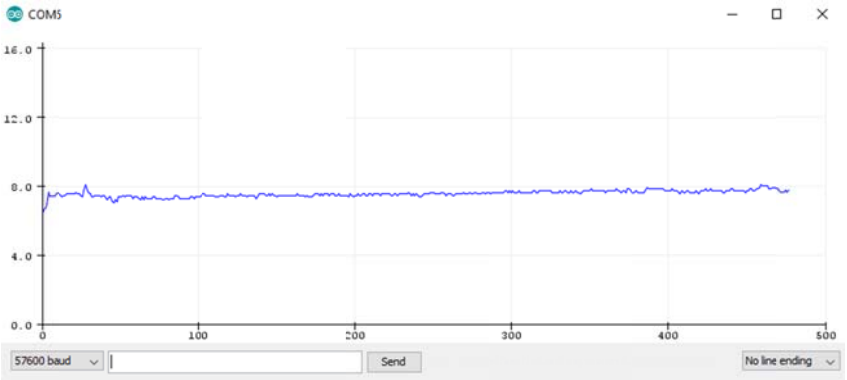


Figure 2.4 – Arduino Serial Plotter results for sustained duration

Conclusion

After calibration, as a software solution we used Arduino and for comparison Logger Lite.
For the code written above, the values of samples of the changing force applied on the hand dynamometer could be seen in Arduino/Serial Monitor for a specified entered character which determines the speed of printed samples (sampling frequency) or releasing and interrupting this series of samples. Graphic interpretation is obtained through Arduino/Serial Plotter. For higher sampling frequency we get more detailed graph.

Literature

1. Vjezba3 Timer interrupt.pdf – www.apeg.me
2. https://www.instructables.com/id/Arduino-Timer-Interrupts/
3. https://www.vernier.com/product/hand-dynamometer/
4. https://www.vernier.com/manuals/hd-bta/
5. https://www.vernier.com/files/manuals/hd-bta/hd-bta.pdf
6. https://en.wikipedia.org/wiki/Dynamometer