Posture monitoring by PIC measurement system based on FSR sensor

Aleksandra Lojaničić, Tijana Đurkić and Đorđe Novaković, *Member, IEEE*

***Abstract*—The aim of this project is to provide a system that detects benching of the back during sitting, the way of connecting and using hardware components needed for building it. First part of this paper includes description of a development board, a sensor and an actuator. The other part gives detailed explanation of a practical work, code implementation, as well as suggestions for further development of this system.**

***Index Terms*—biomedical instrumentation, measurement methods, measurement-acquisition systems in biomedicine**

I. INTRODUCTION

The development of technology and modern lifestyle has led to innovations in computer science and mobile phones that have become vital part of our everyday life. Computers are used in schools, at univeristies, at work, including sitting too long in offices, while mobile phones represent a way of taking a break from daily activities. Using these devices for several hours repeatedly may lead to severe aches in back, shoulders, neck and spinal damage as a consequence of a bad posture while sitting, uncomfortable chairs, strained neck and rounded shoulders and back. This issue is most common among children as they grow up with high-tech gadgets like playstation, xbox while playing video games on PC and using social networks on their phones represent their way of socialising. All of this implies sitting and therefore bad back posture which may cause problems with their health in future. Spinal damage may lead to chronic headaches, anxiety, tiredness and other serious health problems such as tremor and complications in walking. The idea of this project is to design a device that will follow the position of the upper part of spinal cord and produce sound and vibration to warn a user to correct the posture if shoulders and back are bent too much. This system may represent simple, but very effective solution to an issue mentioned above, especially for the people who spend a lot of time in front of computers.

II. THE PRINCIPLE OF THE DEVICE’S OPERATION

FSR sensor is posted on the upper back in order to record flexion of that part of spinal cord and servo motor is attached on hip to simulate vibrations of the device when changes in resistance of FSR sensor occur, caused by back slouching. The whole set-up is illustrated in Figure 1. An arrow 1 is pointed at the FSR sensor attached to the body with an elastic band. Servo motor is marked with an arrow 2.

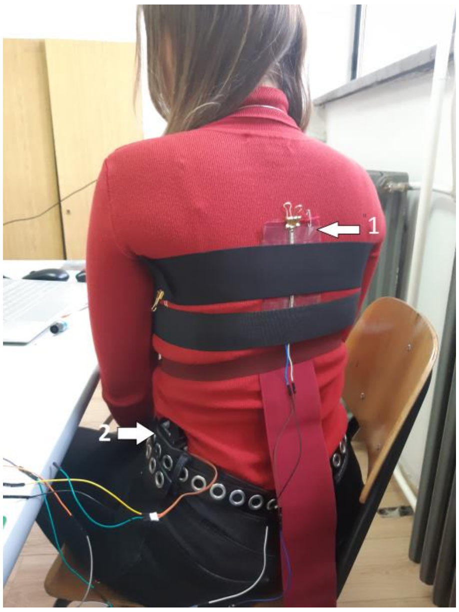

Fig. 1. Illustration of the FSR sensor and servo motor set-up

EasyPIC PRO v7 development board with default PIC18F87K22 microcontroller is used for system formation. Voltage divider made of one fixed and one variable resistor (FSR sensor) is connected to the port A. This way, it is enabled to register changes in resistance of FSR sensor that occur when bending as a change in voltage transmitted to the microcontroller.

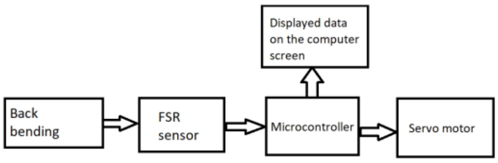

Fig. 2. Block scheme of the device

Aleksandra Lojaničić is with the Faculty of Technical Sciences, University of Novi Sad, Dositeja Obradovica 6, 21000 Novi Sad, Serbia (e-mail: lojanicic.aleksandra@gmail.com).
Tijana Đurkić is with the Faculty of Technical Sciences, University of Novi Sad, Dositeja Obradovica 6, 21000 Novi Sad, Serbia (e-mail: tijana.djurkic@gmail.com).
Đorđe Novaković is with the Faculty of Technical Sciences, University of Novi Sad, Dositeja Obradovica 6, 21000 Novi Sad, Serbia (e-mail: djordjenovakovic@uns.ac.rs).

The threshold of the voltage that indicates slouched back is found experimentally. The person involved in an experiment was bending slowly and voltage values were sent to the computer and observed on the screen. After repeating the experiment for a few times, we agreed which values indicated bad back position. Servo motor is directly attached to the port D, receiving signal for activation from microcontroller. Piezo buzzer that is a part of development board generates audio signal activated when poor posture is detected. The sound was set on the frequency of 1000 Hz because it is unpleasant to human ear and makes a user to correct his/her posture immediately. To achieve proper functionality of the system, corresponding code is implemented in mikroC program.

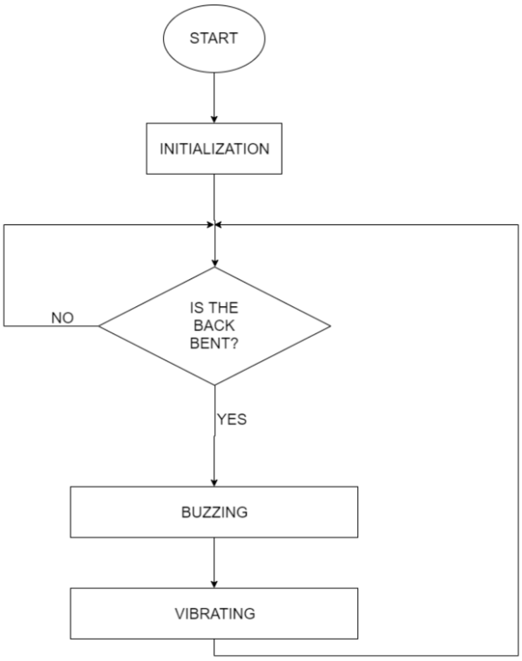

Fig. 3. Algorithm of the system

III. DEVELOPMENT BOARD – EASYPIC PRO V7

EasyPIC PRO v7 development board used in this project is a product of MikroElektronika company and key features and funcionalities will be described in the following paragraph. Power supply from 3.3 V to 5 V can be realised by USB cabel, DC power adapter or additional screw terminals. Default microcontroller of this board is 8-bit PIC18F87K22. Other microcontrollers supported using MCU card placed into the MCU socket are available as well. In addition, 16 MHz crystal oscillator that provides clock frequency, USB communication lines and jumpers for adjusting USB communication and voltage range can be found on the MCU socket. Transfer of data and instructions between a computer and a microcontroller is possible with mikroProg programmer. Input/Output ports are organised in seven groups from A to J and they include tasters, LEDs, pins and switches with three states (enabling pull-up, pull-down or neither of the two mentioned states to be activated). Each mikroBUS socket contains power lines, GND, communication pins: Rx (UART Receive line), Tx (UART Transmit line) and others, as well as few single pins from which PWM (pulse width modulation) was used in this project. Compatible Click Boards can be added to mikroBUS sockets. It is possible to use UART communication via USB or RS-232 connector and achieve connection between devices that support USB and Ethernet communication. Easy PIC PRO provides LCD, analog and digital temperature sensor, reset button, additional GND pins, button press level and piezo buzzer (that was used in this project). Piezo buzzer connects to RB6 pin which can produce PWM signal and buzzing The frequency of the sound is between 2 kHz and 4 kHz.

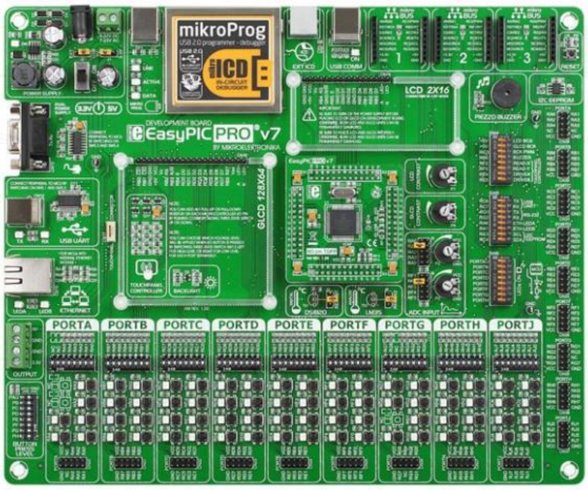

Fig. 4. EasyPIC PRO v7 development board

A. *PIC18F87K22 microcontroller*

PIC18F87K22 is a microcontroller produced by Microchip Company implemented as a part of EasyPIC PRO v7 development board. It is high-performance 80-pin package. Operating voltage range is between 1.8 V and 5.5 V and maximal operating speed is 64 MHz. Flash Program Memory is up to 128 Kbytes, EEPROM 1,024 Bytes while Data Memory is 4 Kbytes and minimum of Erase/Write Cycle Flash Program Memory is 10 000. If the capacity of 128 KB is insufficient, there is an external memory bus of up to 2 MB. Also, this microcontroller incorporates three internal oscillators which can be used, among other things, for adjusting clock specifications such as speed, continued low-speed operation if a failure occurs, a stable reference source... One of significant highlights is 12-bit A/D convertor with 24 channels. LP Watchdog Timer includes a 22-bit prescaler enabling stable extended time-out range in case of prolonged operating. The CTMU is an analog module that measures time difference between impulses generated from different sources and asynchronous pulse generation. Another special feature is an external CCP module that incorporates up to seven Capture/Compare/ PWM modules. It is used for performing several different operations in the same time. In this project, PWM module was of great significance as it generates signal and sets its duration, i.e. duty cycle, based on the frequency requested by a user. Capture module detect rising or falling edges of impulse, therefore duration and Compare module compares values. This microcontroller includes three ECCP (Enhanced CCP) as well as a calendar and real-time clock.

IV. ACTUATOR AND SENSOR

A. Servo motor

As an actuator we used a small, lightweight plastic servo motor. It usually works in the voltage range from 4.8 V to 6 V. It has a small torque of 2.5 kg/cm, but for the implementation of this project it is quite enough because it needs to touch the user weakly when the back is too bent. The angle of rotation ranges from 0° to 180° (90° in each direction). Figure 5 shows the servo motor used in this project. The brown wire connects this actuator to the ground (GND), and the red wire provides power to the servo motor (via the development board). The orange wire sends PWM signals from the microcontroller that drives the propeller. The computer code ensures that impulses are sent at appropriate times, when the back flexion threshold is exceeded.

Fig. 5. Servo motor

B. FSR sensor

To register the bad posture, a FSR sensor was used, which was attached between the user's shoulder blades, along the spine. We chose this sensor because it is cheap and easy to use. The full name-Force Sensitive Resistor, says that when we apply force on this sensor, its resistance changes. Most commonly, carbon-based piezoresistive ink is used for making this sensor. The action of force reduces the distance between the ink particles, and the conduction path and thus the overall resistance. In the case of our project, bending the back bends the FSR sensor, reduces the distance between the particles, and thus the resistance.

Fig. 6. FSR sensor

The voltage divider, implemented on the protoboard, whose electrical diagram is shown in Figure 7, detects the change in resistance of resistor R1, which is an FSR sensor, as a change in voltage signal (Vout) whose value is sent to the computer for further processing. In this way, it is possible for the resistance values of the FSR sensor to be used indirectly in computer code.

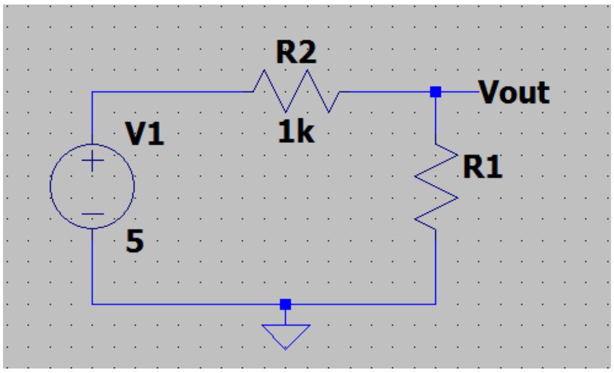

Fig. 7. Voltage divider

V. FIRMWARE

The code for implementing a routine interrupt is shown in Figure 8. The volatile variable is used when a variable can change values due to some external action (e.g. interrupt), regardless of program execution, which happens in this case. If an interrupt has occurred, i.e. if TMR0IF_bit = 1, counter (cnt) is incremented by one. TMR0L is set to 5 because it represents the difference between the maximum value (255) and 250 (2000/23). Using PWM, we set that the period between the occurrence of two signal pulses is 20 ms, and that the interruption occurs every 500 µs, so the counter is reset when it reaches 40 because it means that during one signal period the interrupt happened 40 times. Then the output pin connected to the actuator takes the opposite value. If cnt = duty_cycle, the inversion of the pin value occurs again. The duty cycle represents the part of the interval in which the pulse is located, i.e. the ON part of the period generated by the PWM and determines the movement of the servo motor. Finally, we reset the TMR0 flag.

```
volatile uint8_t cnt = 0;
volatile uint8_t duty_cycle = 2;

void interrupt(){
   if(TMR0IF_bit){
       TMR0L = 0x05;
       cnt++;

       if(cnt == 40){
          cnt = 0;
          LATD0_bit = ~LATD0_bit;
          }
       if(cnt == duty_cycle){
          LATD0_bit = ~LATD0_bit;
          }

       TMR0IF_bit = 0;
     }
}
```

Fig. 8. Implementation of a routine interrupt

The initMain () function primarily specifies the port for the audio output (piezo buzzer) signal and initializes the UART for asynchronous serial communication, as can be seen in Figure 9. A delay of 100 ms is required to synchronize the UART. The interrupt settings were made below in the code. GIE_bit (Global Interrupt Enable) is set to 1 to enable interruption globally. TMR0IE_bit is used to approve a specific interrupt, while TMR0IP_bit is used to select priorities. TRISD0_bit is initialized as an output here. If T08BIT_bit = 1, it is an eight-bit timer, and T0CS_bit = 0 indicates that the internal clock is used which is obtained by prescaling the oscillator. PSA_bit = 0 means that the prescaler is on. T0PS2_bit, T0PS1_bit, T0PS0_bit determine the bits of the TPS prescaler. TMR0ON_bit = 1, turns on the timer.

```
static void initMain(){

  Sound_Init(&PORTB, 6);

  UART1_Init(9600);
  Delay_ms(100);

  UART1_Write_Text("Start1");

  GIE_bit = 1;
  TMR0IE_bit = 1;
  TMR0IP_bit = 1;
  TRISD0_bit = 0;

  T08BIT_bit = 1;
  T0CS_bit = 0;
  PSA_bit = 0;
  T0PS2_bit = 0;
  T0PS1_bit = 1;
  T0PS0_bit = 0;

  TMR0L = 0x05;
  TMR0ON_bit = 1;

}
```

Fig. 9. Initialization of the routine interrupt

In Figure 10, it can be seen that the reading of the voltage value due to the change in the resistance of the FSR sensor is done with the command ADC_Read () which also performs A / D conversion of continuous voltage into the appropriate digital form suitable for processing on a computer. Also, the read value is converted (IntToStr) and the data are displayed or returned to the user.

If the read converted voltage value is less than the threshold that we determined experimentally, i.e. if the back is too bent there will be an audible signal via the piezo buzzer. The first parameter of Sound_Play function determines frequency and the second one represents duration of the sound. In addition, the interrupt allows us that when duty_cycle = 2 the servo motor turns to one side and then when duty_cycle becomes 4 it turns to the other. Numbers 2 and 4 represent final positions of the servo motor (0° and 180°).

```
void main(){
  uint16_t adc_rd;
  char txt[7];
  uint8_t nulta_poz = 0;

  initMain();

  while(1) {
      adc_rd = ADC_Read(0);
      IntToStr(adc_rd,txt);
      ltrim(txt);

      UART1_Write_Text(txt);
      UART1_Write_Text("\r\n");
      Delay_ms(1000);

      if (adc_rd < 1735) {
         Sound_Play(1000, 1000);
           duty_cycle=2;
           Delay_ms(2000);
           duty_cycle = 4;
           Delay_ms(2000);
      }
  }
}
```

Fig. 10. Reading values and activation of the actuator

VI. CONCLUSION

During the implementation of this device, there were problems with placing the sensor on the back, and how to place the sensor to be comfortable for the user. Since the FSR sensor does not have a high sensitivity, it is important that it is held stably and does not move. In future work, these difficulties could be overcome by sewing the sensor on clothing or connecting it to a vest that the user would wear. The device would be more precise if it was attached to different sizes of the vests so that it can be appropriate for both children and adults. Due to the difference in the size and width of the back of the person, a manual adjustment of the threshold could be set, i.e. each user can determine for himself the threshold that needs to be detected. This adjustment could be achieved with connecting it to a simple application on mobile phones via Wireless or Bluetooth. In addition, this app may have an option to track progress over time. Enabling this device to be used by young children is one of the most significant advantages as it could develop habit of sitting correctly from the early age which will prevent scoliosis, neck and head pain etc. in their future. Since these common issues affect a lot of people, it is important to be affordable. With Microchip's nanoWatt technology providing low-power consumption it is possible not to waste too much energy and use batteries as a source of power, reducing the size and the price of the device while prolonging its lifetime. After searching reviews on similar gadgets on the Internet, we came to conclusion that these devices only vibrate and people get used to it so they don't feel it anymore. Innovation in this project is irritating sound which will be considered unpleasant regardless to the time of usage.

VII. REFERENCES

[1] https://www.mikroe.com/easypic-pro-v7

[2] https://download.mikroe.com/documents/full-featured-boards/easy/easypic-pro-v7/easypic-pro-v7-manual-v101.pdf

[3] http://ww1.microchip.com/downloads/en/devicedoc/39960d.pdf

[4] http://ww1.microchip.com/downloads/en/DeviceDoc/ccpwm.pdf

[5] https://learn.adafruit.com/force-sensitive-resistor-fsr/overview

[6] https://components101.com/servo-motor-basics-pinout-datasheet

[7] http://ww1.microchip.com/downloads/en/Market_Communication/nanowatt1jan03.pdf