tinyML on AVR

Assignment no.2

EE-222 Microprocessor Systems

1. Administrivia

1.1. Learning Objectives

The tinyML assignment will enable you to,

- Practice concepts learnt in AVR for solving real world problems
- Understand how we can communicate with embedded devices using serial protocol
- Understand the basic principle of machine learning
- Understand how machine learning models are trained
- Understand how machine learning inference can be done on microcontroller

1.2. Deliverables and Timeline

You are required to deliver and demonstrate a working hardware model of this assignment by April 28, 2023 at 11:59 pm. You are also required to submit a report (max 3 page) on LMS showing the snapshot of your final design and the features implemented. A thorough understanding of the assignment is necessary which will help you in oral viva.

1.3. Marks Distribution

The following marks distribution is tentative and is subjected to change without any prior notice to the students.

Hardware Demo	**Viva**	**Total Marks**
20	10	30

2. Problem Statement

A temperature sensor has been deployed in the environment, which is noisy and also needs to be calibrated. As an embedded system engineer, you will use your knowledge to acquire temperature sensor data, perform noise filtering, apply your machine learning model to calibrate the sensor (training will be done offline against some ground truth) and display the temperature value (Instantaneous, Max, Min) locally as well as send it remotely to a computer terminal for monitoring.

3. Proposed Solution

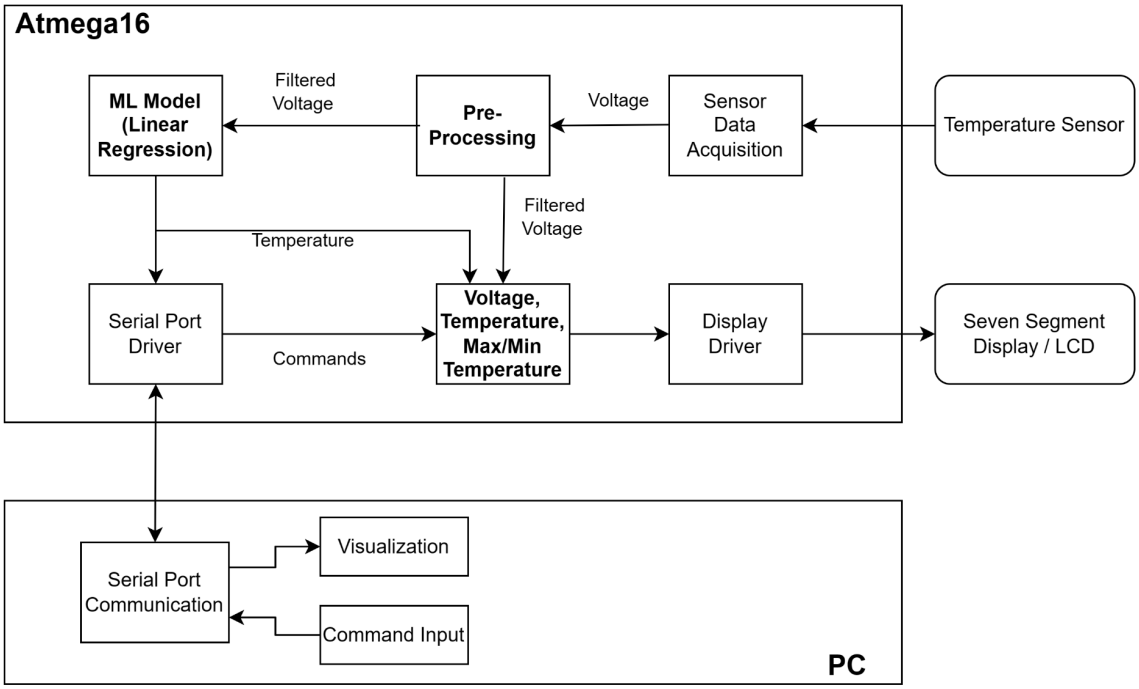

Figure 1. Temperature Sensing using AVR

4. Components

- Arduino Kit with Atmega16 AVR
- LM35 Temperature Sensor
- Seven Segment Display / LCD
- USB-to-Serial Converter
- PC

5. Modules

5.1. Sensor Data Acquisition

In this assignment, you have to use AVR ADC to acquire temperature data. PortA will be used for ADCs. Since we are integrating only one temperature sensor LM35, we will use pin 0 of PortA. For a detailed discussion on temperature acquisition using ADC, please refer to tutorial Sensing Temperature Using AVR - Embedds. Also refer to Lab 8 manual on ADC.

5.2. Pre-Processing

Usually sensor data is noisy and we have to incorporate some pre-processing to filter out noise. This pre-processing is done using a low-pass moving average filter.

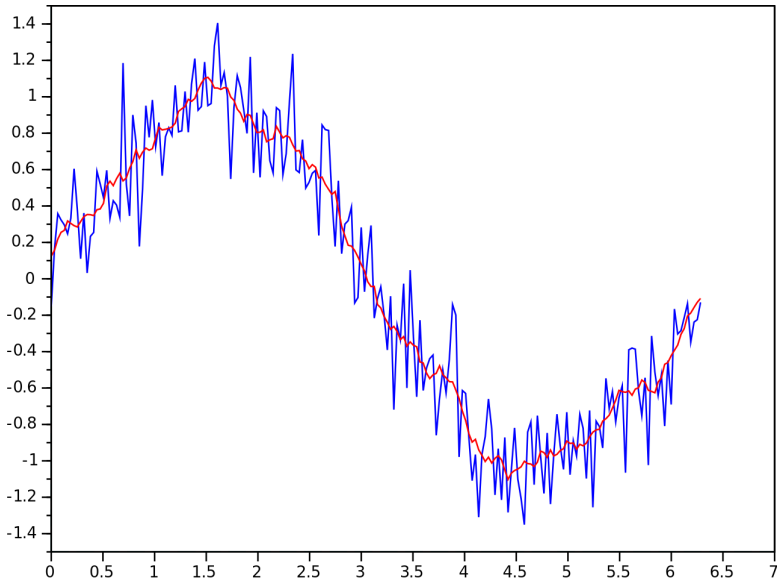

Figure 2. Moving Average Filtering

Y[n] = (x[n] + x[n-1] + x[n-2] + x[n-3])/4;

Here x[n] corresponds to current value (voltage) from the temperature sensor, x[n-1] is previous sample, so an so forth.

5.3. Linear Regression

In Machine Learning, there are usually two types of problems that we solve:

- Regression - we predict the value (real) of dependent variable *e.g. predict value of car given mileage*
- Classification - we train the system to predict class (discrete) *e.g. classify an image to be cat or dog.*

Since this lab is more related to linear regression, we will briefly describe linear regression.

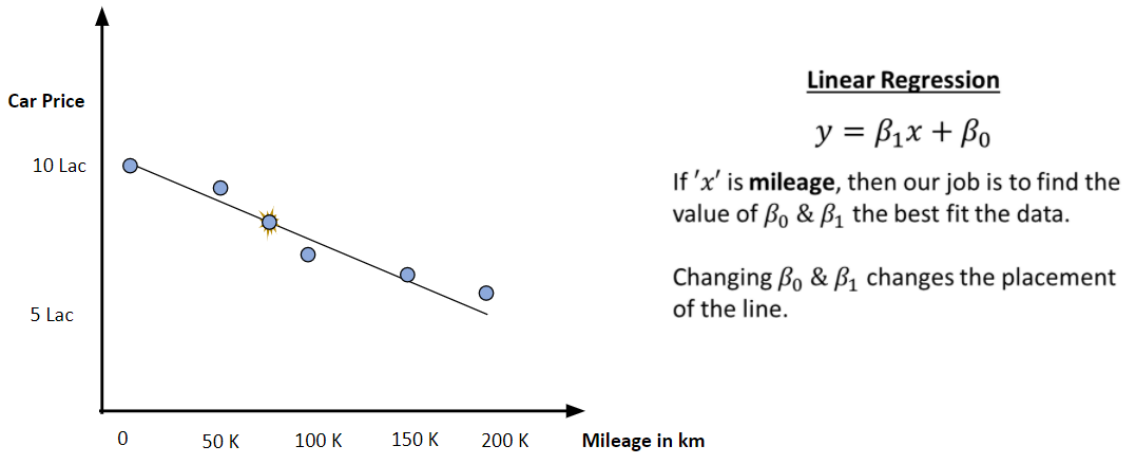

Figure 2. Linear Regression : Estimation of Car Price

This idea is to collect values of *x* (Mileage in km) and *y* (Car Price) for a few training samples and then estimate the parameters β_0 and β_1. Once we have the parameters we can estimate any car price given any mileage.

In our case, we have to note down the values of x (temperature from LM35) and y (temperature from thermometer) for some values. Since x will be noisy due to inaccurate LM35 sensor, once fitted using the linear regression will provide value close to the thermometer.

Note: You have to convert voltage coming from LM35 into temperature first Sensing Temperature Using AVR - Embedds.

5.4. Max, Min Computation

Once we have obtained temperature (after regression), we need to continuously find out the minimum and maximum temperature value. The command will be received from the serial port as to which value should be displayed on a seven segment display (nominal, minimum, maximum).

5.5. Serial Port Communication

In order to communicate with PCs, we can use the AVR serial port. Lab 7 of this course is about Serial Communication. Please follow lab instructions to carry out serial communication with the PC. On the PC side, Python should be used for opening serial port and sending/receiving data to and from AVR microcontroller Python - Python Serial Communication (pyserial) (devtut.github.io).

5.6. Graphical User Interface

A time series graph needs to be displayed on a PC using Python indicating temperature value over a period of time. Use matplotlib in Python to display graph. Also there should be a selection whether we need to display minimum/maximum/nominal value of temperature on the Seven Segment Display. This command should be sent to AVR microcontroller using serial port.

6. Work BreakDown Structure

6.1. Data Acquisition and Display

You need to configure a timer to acquire data from sensor after every second. After every interrupt from timer, you will acquire data from ADC and display it on a Seven Segment Display.

6.2. Pre-Processing

You need to extend 6.1 by adding pre-processing block whereby you have to implement a 4-tap moving average filter. The filtered output (voltage) will be displayed on a Seven Segment Display.

6.3. Min/Max Block

You need to extend 6.2 whereby the pre-processed output will be passed on to a min/max computation block and based on fixed setting within code, either min/max or nominal value (voltage) will be displayed on Seven Segment Display.

6.4. Temperature Calibration

You need to extend 6.3 whereby the pre-processed voltage will be then calibrated using formulas mentioned in https://embedds.com/sensing-temperature-using-avr/ and a temperature value will be obtained. This temperature value will be passed on to the min/max computation module and also will be displayed on Seven Segment Display.

6.5. Serial Port (AVR side)

You need to extend 6.4 by configuring serial port on AVR microcontroller and then sending data to PC and getting it displayed on teraterm software within PC. Refer to Lab Manual 7.

6.6. Serial Port (AVR + PC)

You need to extend 6.5 by configuring serial port on PC using Python and displaying data coming from serial port first on console and then using matplotlib to display in graph.

6.7. Commands

You need to extend 6.6 to send commands from PC to AVR over serial port for displaying either min/max/nominal value of temperature on Seven Segment Display

6.8. Training ML Model

You need to modify 6.4 whereby you will note down a number of readings of different temperature settings of room and noting down readings of voltage on Seven Segment Display and also temperature value from some digital thermometer (we need to figure out this). We should note down at least 20 readings of x (sensor data voltage), y (temperatue reading of some digital thermometer). We will use ScikitLearn library in Python for linear regression to estimate the parameters as mentioned in Section 5.3. You need to use these parameters to implement equation of Section 5.3 whereby we obtain temperature using the machine learning model.

7. Conclusion

The purpose of this assignment is to apply the concepts learnt in Microprocessor Systems (microcontroller part) including sensor data acquisition, timers, interrupts, serial port communication and seven segment display. This will help you to implement any kind of embedded system using microcontrollers.