School of Electrical Engineering and Computer Science

CptS466/566: Embedded Systems

Fall 2018

Final Project - Project 6 (P6)

Code & Report Due: 12/07/2018 @ 11:59pm (Blackboard)

Demo Due: 12/07/2018 between 2pm and 4pm (Dana 118A)

1. Preparation

You will need

- LaunchPad,
- Bluetooth module
- TMP36-Analog Temperature sensor,
- Photo cell (CdS photoresistor): http://epsl.eecs.wsu.edu/wp-content/uploads/2018/11/photocells.pdf
- Breadboard
- Some wires.

You will use a terminal application such as

- CoolTerm: http://freeware.the-meiers.org

which can perform serial port communication on the PC for data collection and visualization.

You will also need to use a Machine Learning package such as

- Weka: https://www.cs.waikato.ac.nz/ml/weka/downloading.html

for machine learning algorithm training.

2. Project Description

The goal of this project is to explore if we can use sensors such as temperature and photocell sensors to measure the amount of screen time. Screen time is the time you spend watching TV or DVDs, using computers, playing video or hand-held computer games, or using tablets or smartphones. The system that you will be developing can be potentially (not as part of this assignment) built into a wearable device with sensors embedded in eye glasses, earing, etc. to capture person's exposure to computer, TV, etc.

Your approach to developing a screen time monitoring system should be a machine learning approach that uses supervised learning to classify signal readings into 'screen' 'non-screen' events. As your system continuously detects the 'screen' versus 'non-screen' events, it can keep track of the time that that system identifies 'screen' events and therefore computes the total amount of screen time for its user.

Many details of the project are intentionally left to you to decide but all of these are required to be discussed in your report.

There are four different phases for this project. First, you need to develop a code which reads temperature and photocell sensors and store the signal readings. Second, you need to collect data for different scenarios. Third, you perform offline machine learning on the collected data. Fourth, you will implement your selected machine learning approach on the board. Therefore, it would perform real-time machine learning.

3. Experiments

Building upon your previous projects, write software code for collecting sensor data in different scenarios. Ensure that you collect at least one minute of sensor data for each unique scenario. You are required to design an experiment that captures sufficient cases such as indoor, outdoor, room, ambient-light-on, ambient-light-off. For each scenario, consider multiple (at least two) screen devices such as TV and Laptop. In each scenario, collect sensor data for both classes

- ‘screen’ class: sensor data are collected when the sensors are placed in close proximity of a screen device such as TV or Laptop computer.
- ‘non-screen’ class: sensor data are collected in absence of a screen device.

The sensor data need to be collected wirelessly (e.g., using Bluetooth) and gathered in separated files on a computer. Each file will need to have a class label associated with it. This is often referred to ground truth labeling of the sensor data or sensor data annotation.

4. Machine Learning

Use the sensor data that you collect to learn at least two classification models. You may try more than two classification models and choose the two best algorithms to include in your report. For each classification algorithm, you will need to learn the actual model and also compute and report its performance.

As discussed in the class, the process of machine learning algorithm development involves extracting features from sensor data. Initially, extract an exhaustive set of statistical features from a moving window over the sensor signals. This part needs to be done in a separate software tool than Weka or other machine learning packages. The output will be a training dataset with all features and a class label for each instance/observation.

Use a machine learning tool to perform feature selection prior to classification algorithm training.

In your report, discuss performance of your algorithm (e.g., accuracy, precision, recall). Also, discuss how many features and what features you have initially extracted from the sensor signals. How many of those features did you decide to include in your final classification model?

Additionally, discuss how you validated the trained classifier. Different approaches can be used such as cross-validation, leave-one-out, etc. Include results for various validation scenarios. Note that this process of algorithm validation is done off-line in the machine learning tool (e.g., Weka).

5. Real-Time Screen-Time Monitoring

Choose one of the machine learning classifiers that you trained and implement it on the microcontroller for real-time screen time monitoring. As you extract (selected/final) features from the sensor signals, your implementation needs to decide if the current event is a ‘screen’ event or ‘non-screen’ and these labels need to be transmitted in (near) real-time to the base-station (e.g., your computer).

6. Report

In a separate written document (in Word or PDF), compile sections A through D as follows:

A: System Design. Briefly, in one paragraph, describe your high level design. Discuss any specific observations that you had, any difficulties that you ran into while designing or testing the system. Discuss initial sampling frequency that you used to collect sensor data.

B: *Experiment*. Discuss how you conducted the experiments, in what environments, and any challenges associated with designing a realistic experiment.

C: Results. Include results of at least two classifiers that you tried for event classification. Include at least accuracy, precision, and recall as your performance measures. How do you think you could experimentally minimize the sampling frequency that need to be used in your final implementation for real-time screen time monitoring? Which one of the sensors (temperature / photo cell) do you think contributed most to the ‘screen’ event classification?

Discuss performance of your algorithm (e.g., accuracy, precision, recall). Also, discuss how many features and what features you have initially extracted from the sensor signals. How many of those features did you decide to include in your final classification model?

Additionally, discuss how you validated the trained classifier. Different approaches can be used such as cross-validation, leave-one-out, etc. Include results for various validation scenarios. Note that this process of algorithm validation is done off-line in the machine learning tool (e.g., Weka).

D: Discussions. Discuss your process of feature extraction, feature selection, and machine learning algorithm training. What feature did you extract? What features were eventually selected for inclusion in your classifier?

Discuss your results for real-time screen-time monitoring. Did your real-time approach worked in classifying the events correctly? How many scenarios did you use to test your real-time monitoring system? Describe those scenarios. Is your system performing better in certain cases/scenarios than others?

Do you think your designed system was successful for the targeted application (screen time monitoring)? If successful, what do you think was the main reason that your design was successful? If unsuccessful, discuss what could have been done differently? Did you need to use a different type of sensor? Or did you need a better algorithmic approach? Or do you believe that the experiment did not include sufficient amount of data that captures all realistic scenarios and therefore distortion of sensor data for training and test data was not similar?

7. What to submit?

Submit a .zip file with the following content:

- Your well commented C source code implementing the application described above. This includes both data collection software and real-time monitoring software.
- A report that documents your development process, explanation of the experiments conducted, machine learning design, learned classifiers, discussion of your observations, and a discussion of your design decisions.
- All experimental data collected. You can include these raw data in plain text files with each file having a representative file-name. For example, for one minute of sensor data collected in indoor environment and in front of laptop you can choose a file name such as 'indoor-laptop-screen.txt'.

8. Grading

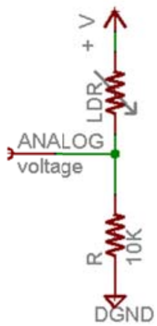
Assume that the whole assignment is worth **200** points.

- 35 pts for well commented and correct C source code for data collection.
- 45 pts for well commented and correct C source code for real-time screen time monitoring.
- 30 pts for experimental data files included in your submission.
- 20 pts for reporting offline machine learning analysis on the collected data (add as a separate section to your report).
- 20 pts for showing a demo of your system for real-time screen time monitoring.
- 50 pts for a well written report that captures all questions and discussion points mentioned previously in this assignment (see section 'Report' of this project assignment).

9. Appendix A – Example System Schematic

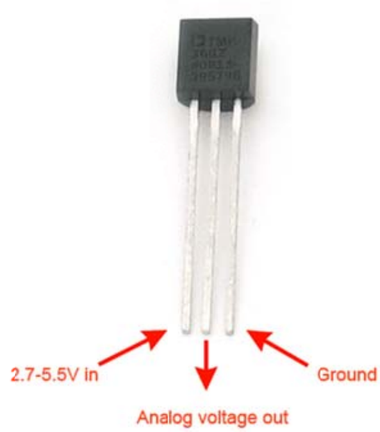
You can interface the Photocell sensor based on the below schematic. Note that LDR is photocell sensor. This method does not provide linear voltage with respect to brightness. Also, each sensor will be different. As the light level increases, the analog voltage goes up even though the resistance goes down.

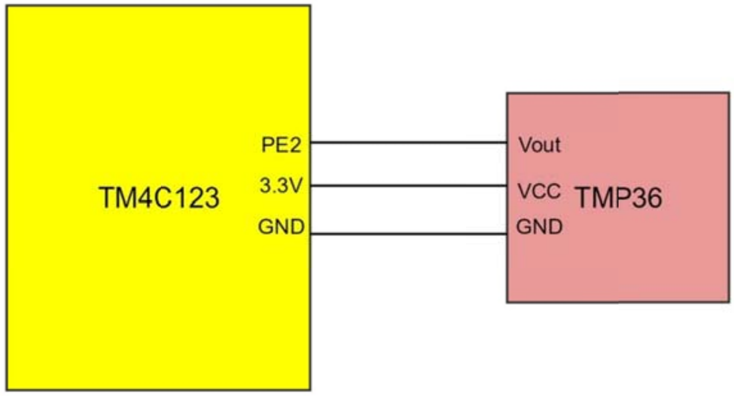
Vo = Vcc (R / (R + Photocell))



10. Appendix B – Example System Schematic

You can interface the temperature sensor based on the below schematic. PE2 is ADC channel 1 which is a potential pin that you can use.




TM4C123
PE2
3.3V
GND
Vout
VCC
GND
TMP36