Keep in Touch

EECS 149 PROJECT
12/17/14

NIRANJAN KUMAR
AMY LIAO
SIDDARTH SEN
ERIC TU

Project Motivation

- Problem:
  - Current touch type learning software has no way to enforce correct finger usage.
  - Incorrect finger usage can lead to slower typing times as well as hand and arm pain.
- Proposal:
  - Develop a Smart-Glove to provide real-time feedback on finger usage
- Design Criteria:
  - Able to detect > 1060 keystrokes/min (fastest typing speed)
  - Force Range = 0.25-0.85N (Gerard 1999)
  - Sensitivity (true positive) > 90%
  - Specificity (true negative)> 90%

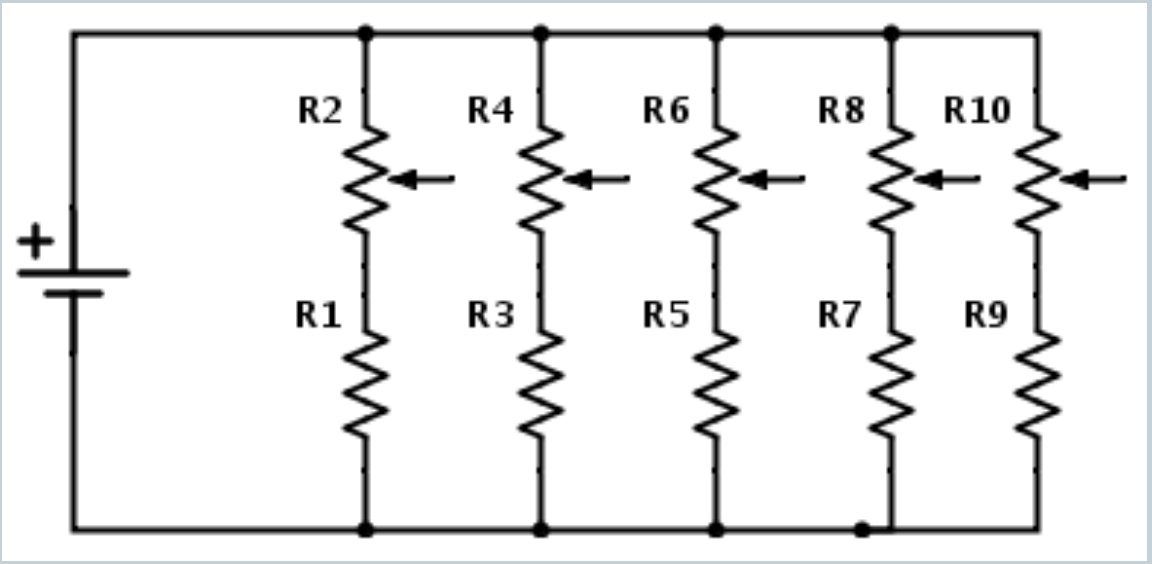
System Design

Smart glove with force sensors on each finger

Freescale mbed system conducts basic thresholding processes

Software compares pressure data with keystroke data to determine finger presses. GUI provides feedback for the learn-to-type course.

Force Sensor



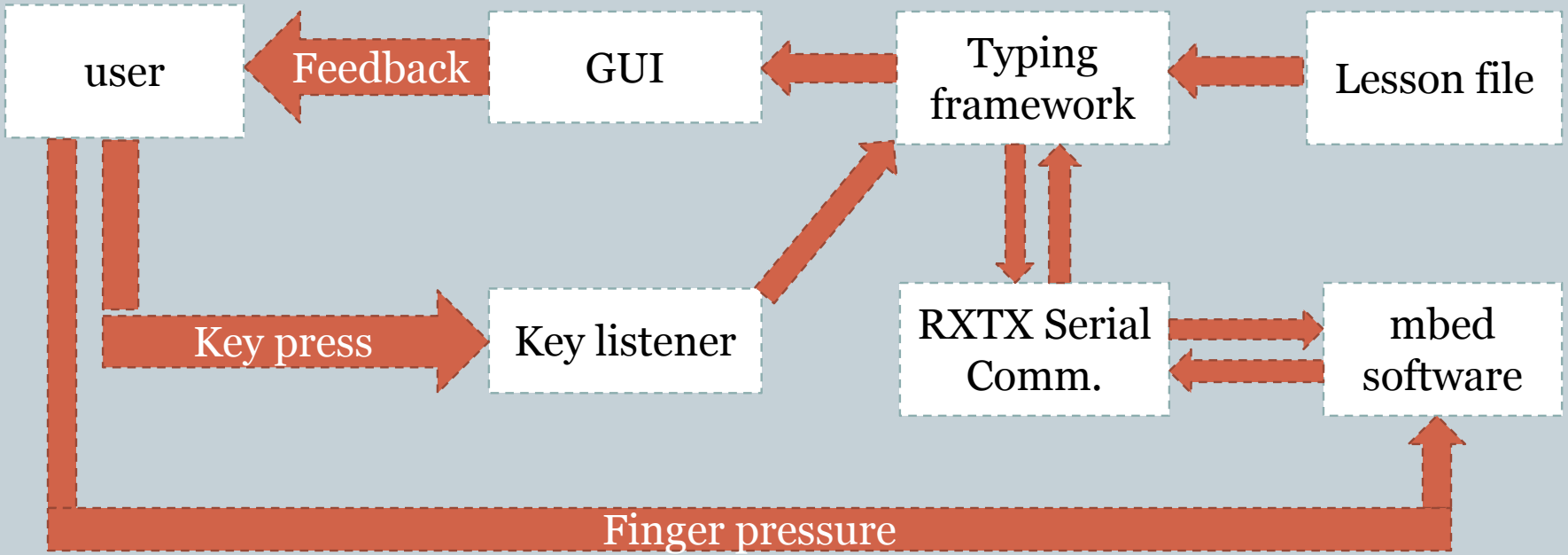
Sensitivity =
0.1N – 10N

Resistance =
1MΩ-2.5kΩ

Non-actuated
resistance = 10MΩ

Linear Range = 0.2N-100N

Software Architecture



Timing and Synchronization

- Importance of Timing in Timestamps
  - Need to distinguish between keystrokes
  - Fastest typer generates 1060 keystrokes/min = 17.67 keystrokes/sec
    - Nyquist's Theorem: must sample at twice that = 35.33 keystrokes/sec
    - The mbed code collects data at 40 samples/sec
  - Ticker object from mbed.h library
    - Creates a timed interrupt at desired frequency
    - Ticker object's attach() method is used to specify ISR
    - Our ISR sends finger data to the PC 40 times per second
- Synchronization:
  - PC sends signal to mbed to start Ticker
  - Millisecond precision required, so serial port delays are okay.
  - 2 mbeds implies 2 data streams need to be processed
    - Need to be efficiently interleaved in chronological order

Inference Problem

- Formalized problem:
  - X = finger used by user + timestamp recorded on MBED
  - Y = key pressed on computer + timestamp record on PC
  - Goal: find P(X|Y)
    - Assume a uniform a distribution on X (user is equally likely to have selected any key)
    - Solving for the Maximum a Posteriori estimate of X given Y simplifies to find the X which maximizes P(Y|X).
      - P(Y|X) can be constructed ourselves using some assumptions
        - Given the timestamp of X, the timestamp of Y is likely to very close that of X
        - Given the finger used, it is unlikely that a key very far from that finger was pressed. (This assumption can be used to break ties or filter out nonsensical results.)

Next Steps

- Improved force sensor
  - More flexible
  - Bigger surface area to detect key presses
- Incorporate a calibration mode to account for keyboard setup
- Add a non-uniform prior to better predict finger usage

Project Demo

- https://www.youtube.com/watch?v=Wm44T_qKRSc&list=UUjNLSIb242IxZ-J8Db39vYQ