Accelerometer Pen

CRITICAL DESIGN

MEET THE TEAM

Kyle Buchanan

Theodore Pham

James Chang

Motivation

- Technically challenging
- Good way to create backup notes
- Could be utilized by groups such as SSDS
- Lots of room to expand or narrow scope

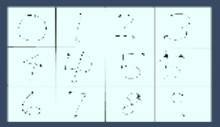
Functionality

- Recognition of shapes, symbols, and numbers on a flat surface
- Wireless communication of written character from pen to board
- Incorporates an accelerometer to measure the acceleration of the pen - Built in non-fillable pen cartridge for visibility of written work
- Displays user written input on a LCD screen
- Uses flash memory to store the character templates
- User click to activate writing sensor
- Power button to turn the pen on/off

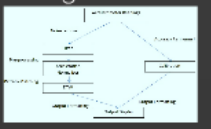
Pattern Recognition

- Selected Algorithm is Dynamic Time Warping
  - Finds optimal matches between time series
  - Used in other pattern matching applications
  - Uses dynamic programming to find ideal paths

Pattern Recognition



Design: Software



Design Calculations

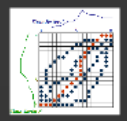
- Data Filtering: 44 FLOPS
- DTW algorithm:
  - time cost: O(n^2)
  - space cost: O(n)
- Battery Runtime on 8 h wake/ 4 h sleep cycles
  - Coin Cell Battery: ~3.00 h
  - Rechargeable AA:~29.40 h
- I/O rates: 36 bits/interval
- Xbee modules: 625 bits per 2.5 ms
- Flash memory required per character: ~3.5 kB
- Onboard Flash: 4 MB ~= 1165 patterns

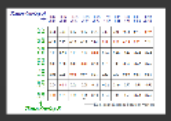
Challenges Thus Far

- The Bluetooth module could not act as a Master to poll data from the accelerometer without a microcontroller so we decided to use wireless Xbee module instead.
- Trying to fit all components into a reasonable sized pen body.
- Accurately prediction of accelerometer output data

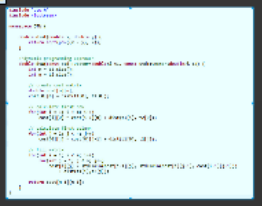
Design: graphical hierarchy of source code

Design: Software





Awesome Code



Optional Features

- Extra button to change pen functionality to perform basic arithmetic
- Option to allow pen to be user dependent (calibrates pen according to users initial written input pattern)
- Ability to recognize alphabets (in future - handwriting)
- Design a GUI for character display on the computer
- Sensor activated without use of a button trigger or pressure sensor
- Incorporate Bluetooth and integrate this new design so that users can connect the pen to their smartphones and generate outputs to an app

Components

- Implementation on DE2 Board:
  - Xbee transceiver module
  - USB connection to computer
- Implementation on Pen Unit:
  - Xbee transceiver module
  - Accelerometer
  - Power Source
  - Switches (Trigger, Power)

Results of Unit and Integration Testing

DTW algorithm test
- Test input with 4095 data points
- runs in 0.172s using an i7@3.4GHz

Application Notes

Although we do not have application notes of our own yet, we are using an Xbee module from a past project on 2012.

Design: Hardware

Accelerometer Pen

CRITICAL DESIGN

MEET THE TEAM

Kyle Buchanan

Theodore Pham

James Chang

Motivation

- Technically challenging
- Good way to create backup notes
- Could be utilized by groups such as SSDS
- Lots of room to expand or narrow scope

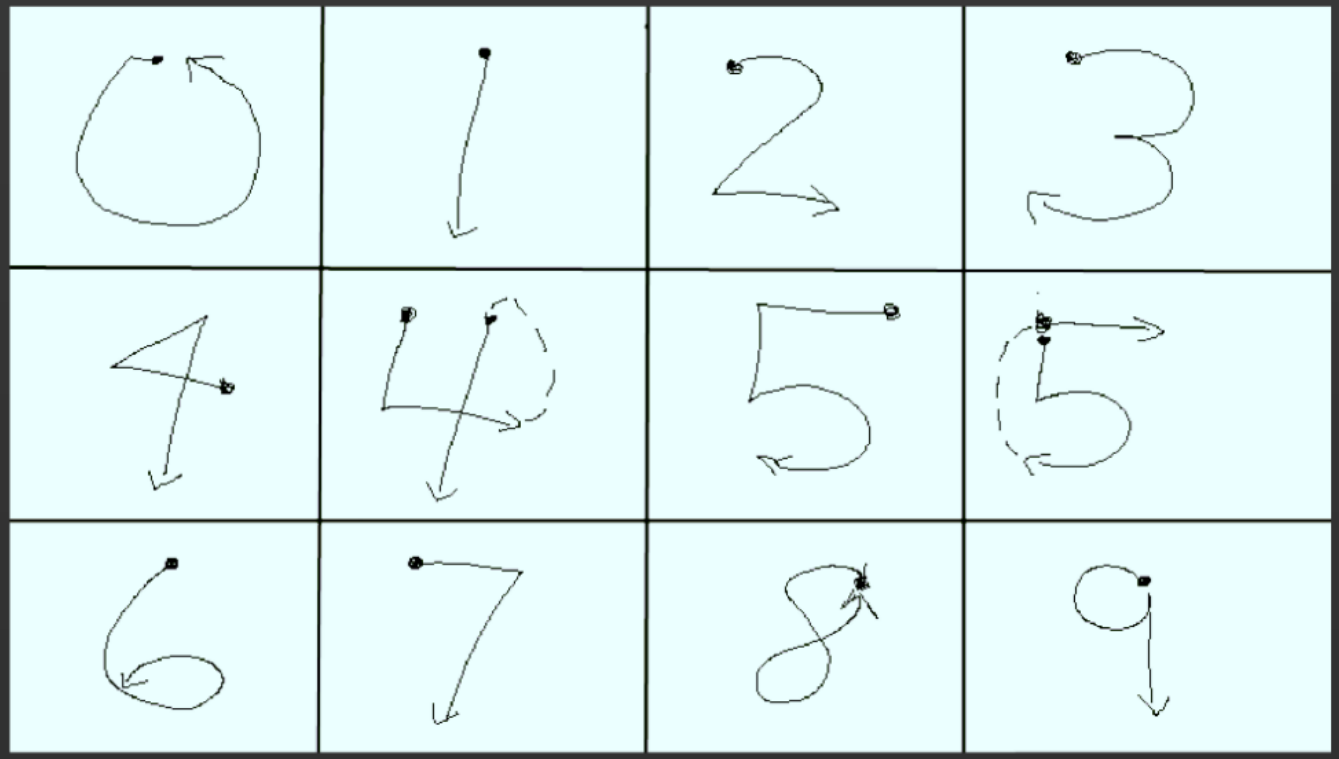
Functionality

- Recognition of shapes, symbols, and numbers on a flat surface
- Wireless communication of written character from pen to board
- Incorporates an accelerometer to measure the acceleration of the pen - Built in non-fillable pen cartridge for visibility of written work
- Displays user written input on a LCD screen
- Uses flash memory to store the character templates
- User click to activate writing sensor
- Power button to turn the pen on/off

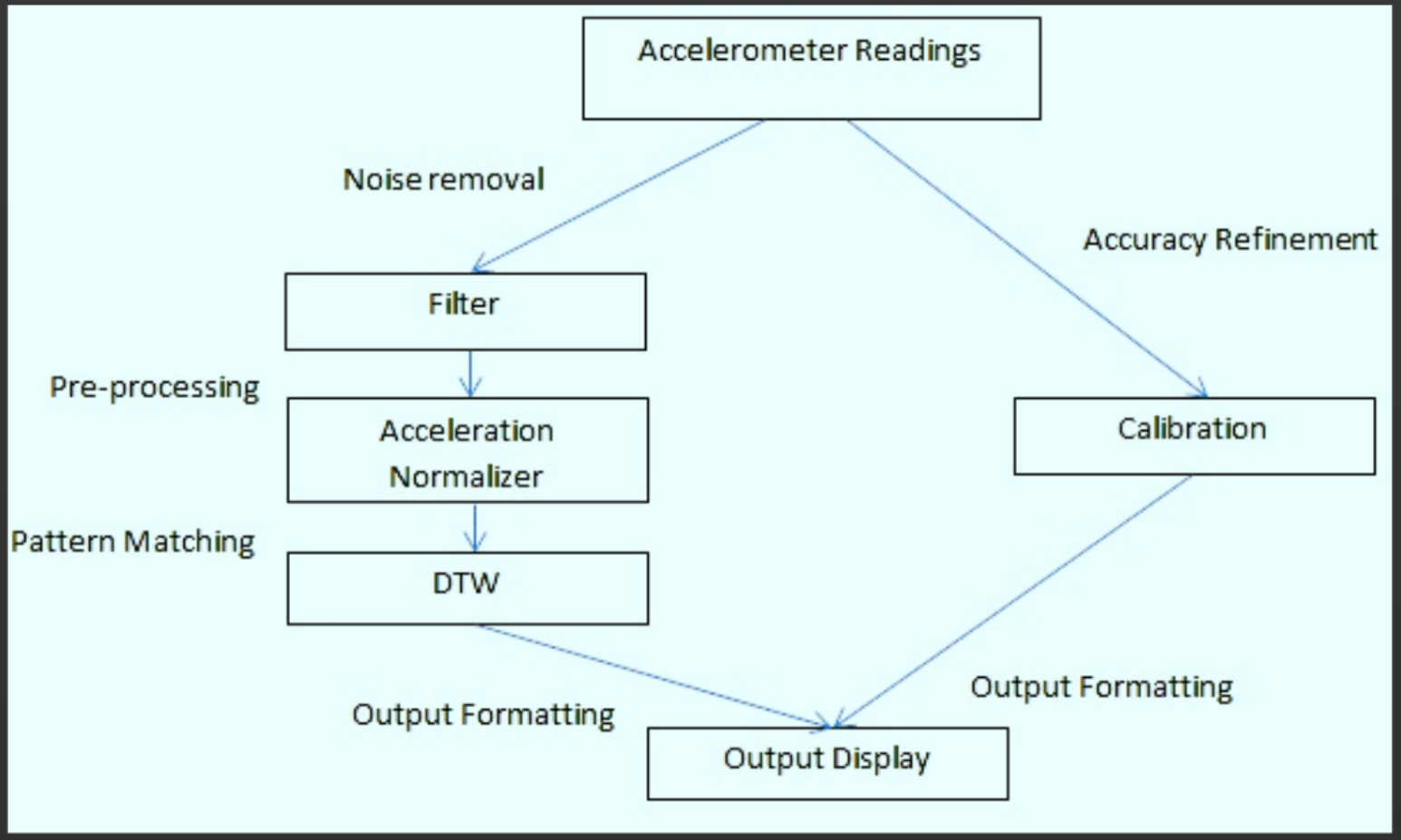
Pattern Recognition

- Selected Algorithm is Dynamic Time Warping
  - Finds optimal matches between time series
  - Used in other pattern matching applications
  - Uses dynamic programming to find ideal paths

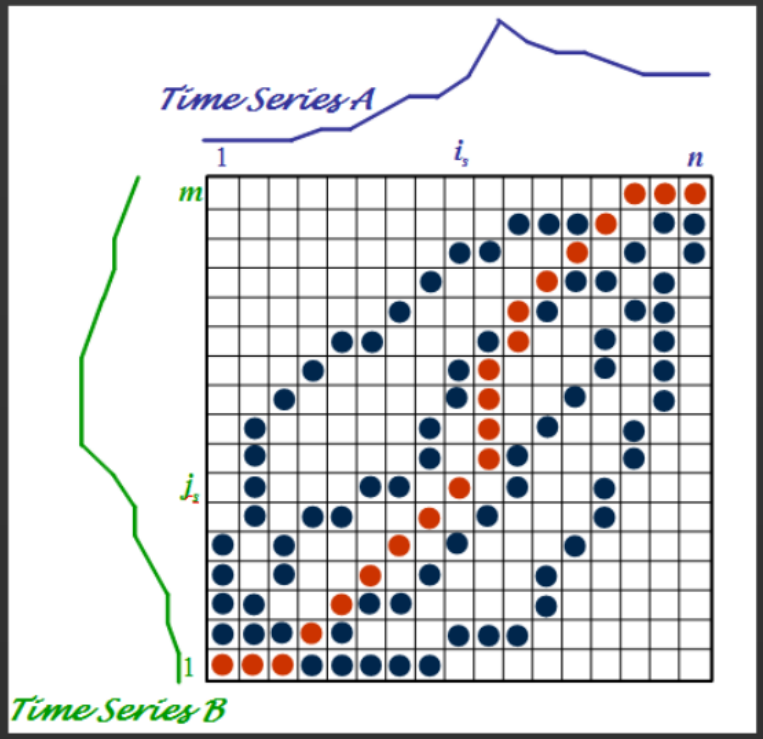
Pattern Recognition



Design: Software



Design: Software



Time Series A

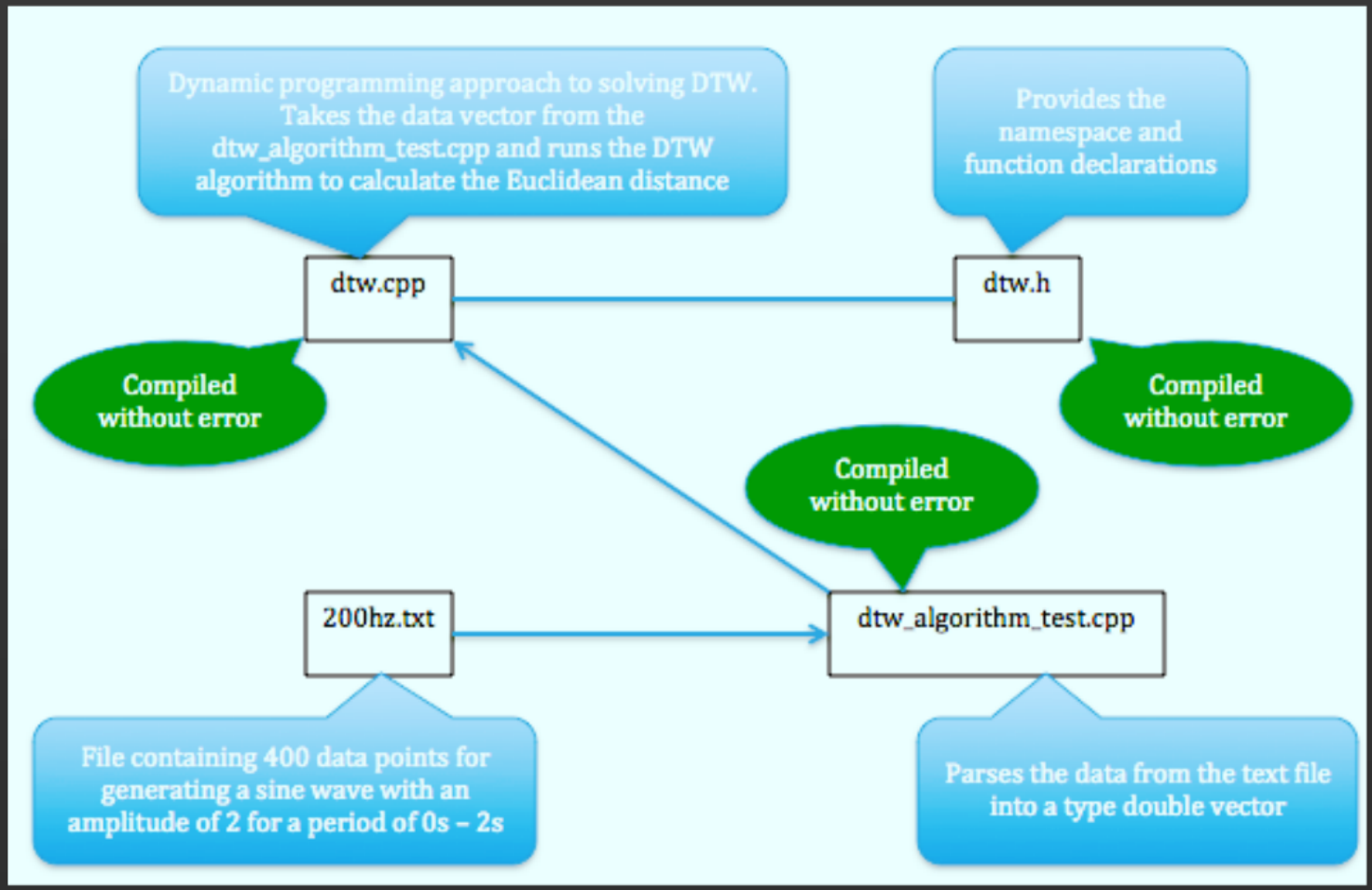
-0.87	-0.84	-0.85	-0.82	-0.23	1.95	1.36	0.60	0.0	-0.29
-0.88	-0.91	-0.84	-0.82	-0.24	1.92	1.41	0.51	0.03	-0.18

1.94 1.97	0.51	0.51	0.49	0.49	0.35	0.17	0.21	0.33	0.41	0.49
0.77 0.74	0.27	0.27	0.26	0.25	0.16	0.18	0.23	0.25	0.31	0.68
-0.17 -0.32	0.13	0.13	0.13	0.12	0.08	0.26	0.40	0.47	0.49	0.49
-0.58 -0.63	0.08	0.08	0.08	0.08	0.10	0.31	0.47	0.57	0.62	0.65
-0.71 -0.68	0.06	0.06	0.06	0.07	0.11	0.32	0.50	0.60	0.65	0.68
-0.65 -0.62	0.04	0.04	0.06	0.08	0.11	0.32	0.49	0.59	0.64	0.66
-0.60 -0.46	0.02	0.05	0.08	0.11	0.13	0.34	0.49	0.58	0.63	0.66

Euclidean distance between vectors

Time Series B

Design: graphical hierarchy of source code



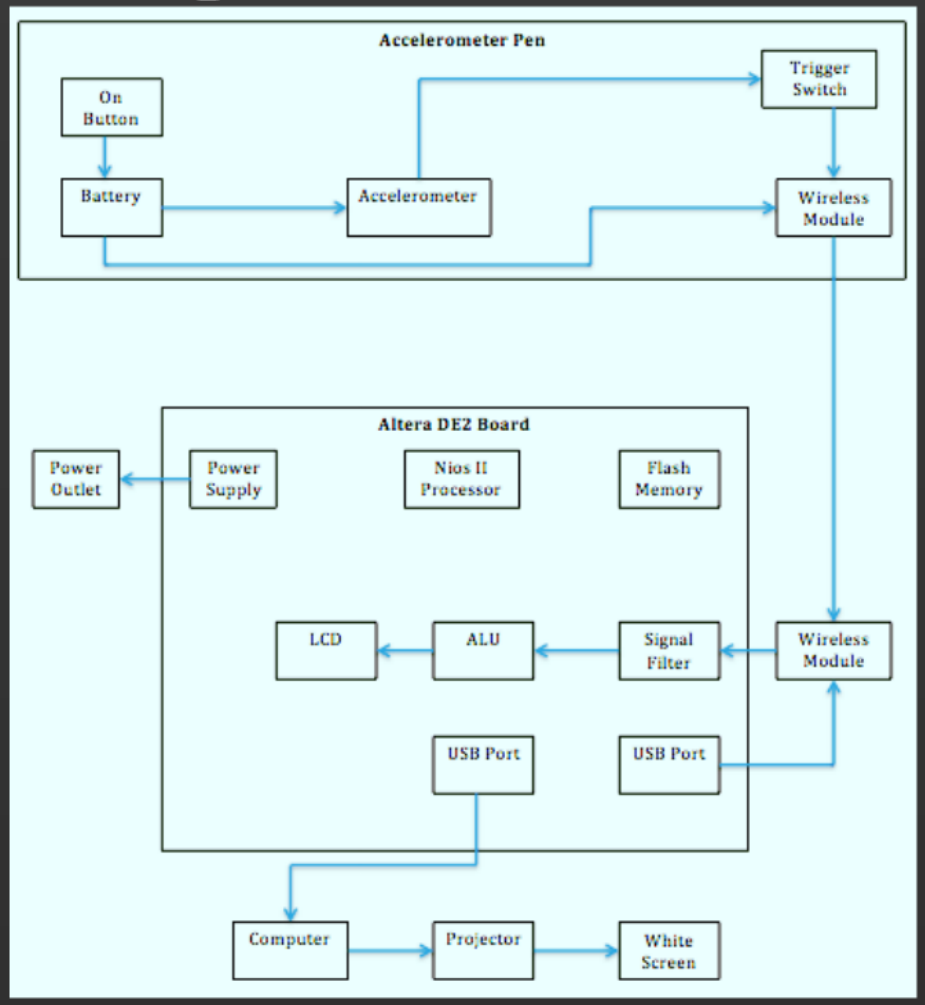
Awesome Code

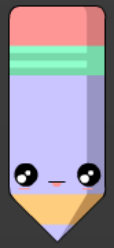
```cpp
#include "dtw.h"
#include <iostream>

namespace DTW {

    double dist(double x, double y) {
        return sqrt(pow((x - y), 2));
    }

    //dynamic programming approach
    double dtw(const std::vector<double>& t1, const std::vector<double>& t2) {
        int m = t1.size();
        int n = t2.size();

        // create cost matrix
        double cost[m][n];
        cost[0][0] = dist(t1[0], t2[0]);

        // calculate first row
        for(int i = 1; i < m; i++)
            cost[i][0] = cost[i-1][0] + dist(t1[i], t2[0]);

        // calculate first column
        for(int j = 1; j < n; j++)
            cost[0][j] = cost[0][j-1] + dist(t1[0], t2[j]);

        // fill matrix
        for(int i = 1; i < m; i++)
            for(int j = 1; j < n; j++)
                cost[i][j] = std::min(cost[i-1][j], std::min(cost[i][j-1], cost[i-1][j-1]))
                    + dist(t1[i],t2[j]);

        return cost[m-1][n-1];
    }
}
```

Components

- Implementation on DE2 Board:
  - Xbee transceiver module
  - USB connection to computer
- Implementation on Pen Unit:
  - XBee transceiver module
  - Accelerometer
  - Power Source
  - Switches (Trigger, Power)

Design: Hardware





Design Calculations

- Data Filtering: 44 FLOPS
- DTW algorithm:
  - time cost: O(n~2)
  - space cost: O(n)
- Battery Runtime on 8 h wake/ 4 h sleep cycles
  - Coin Cell Battery: ~3.00 h
  - Rechargeable AA:~29.40 h
- I/O rates: 36 bits/interval
- Xbee modules: 625 bits per 2.5 ms
- Flash memory required per character: ~3.5 kB
- Onboard Flash: 4 MB ~= 1165 patterns

Optional Features

- Extra button to change pen functionality to perform basic arithmetic
- Option to allow pen to be user dependent (calibrates pen according to users initial written input pattern)
- Ability to recognize alphabets (in future - handwriting)
- Design a GUI for character display on the computer
- Sensor activated without use of a button trigger or pressure sensor
- Incorporate Bluetooth and integrate this new design so that users can connect the pen to their smartphones and generate outputs to an app

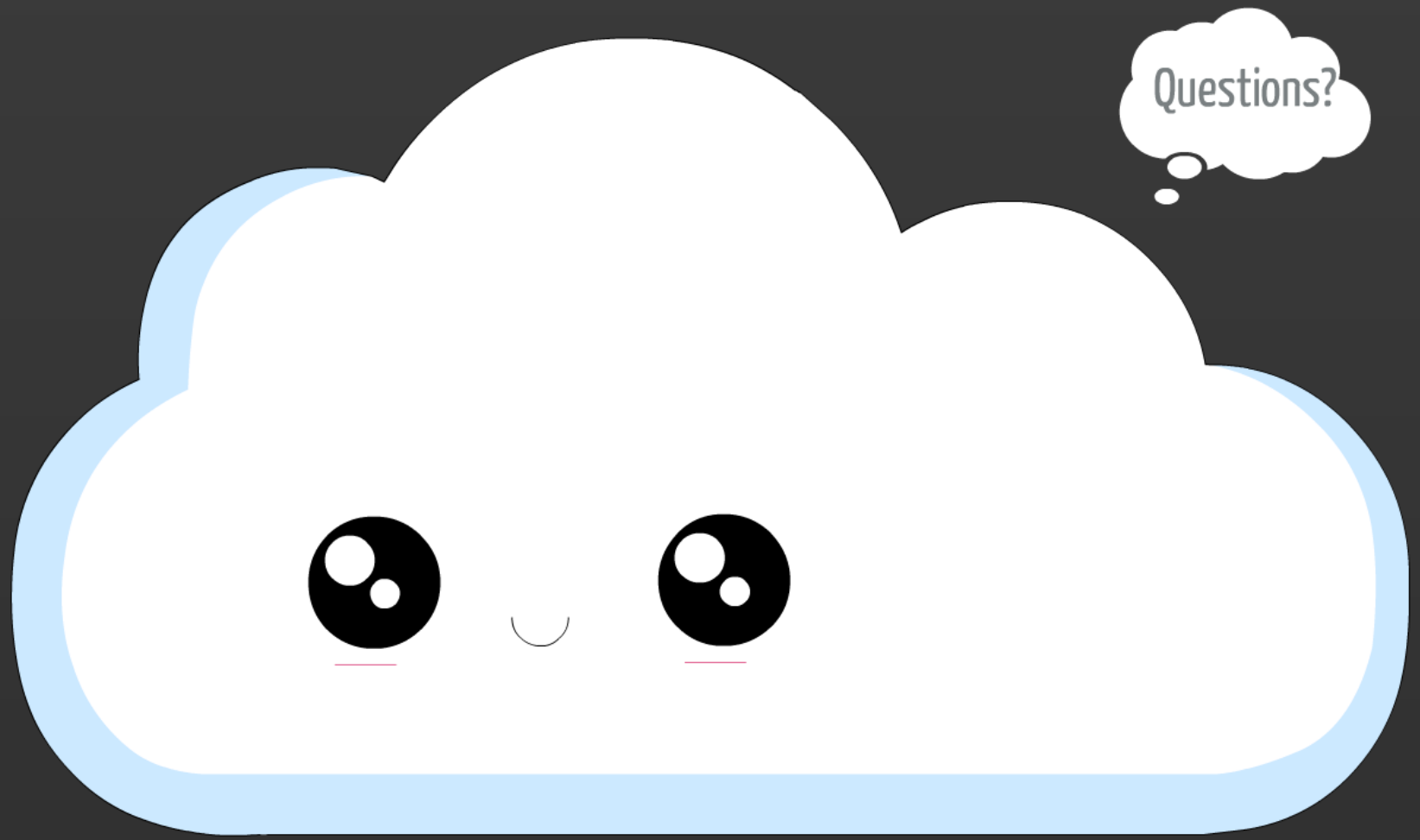
Results of Unit and Integration Testing

DTW algorithm test

- Test input with 400 data points
- runs in 0.171s using an i7@3.4GHz

Challenges Thus Far

- The Bluetooth module could not act as a Master to poll data from the accelerometer without a microcontroller so we decided to use wireless Xbee module instead.
- Trying to fit all components into a reasonable sized pen body.
- Accurately prediction of accelerometer output data



Accelerometer Pen
CRITICAL DESIGN

MEET THE TEAM

Kyle Buchanan

Theodore Pham

James Chang

Motivation

- Technically challenging
- Good way to create backup notes
- Could be utilized by groups such as SSDS
- Lots of room to expand or narrow scope

Functionality

- Recognition of shapes, symbols, and numbers on a flat surface
- Wireless communication of written character from pen to board
- Incorporates an accelerometer to measure the acceleration of the pen - Built in non-fillable pen cartridge for visibility of written work
- Displays user written input on a LCD screen
- Uses flash memory to store the character templates
- User click to activate writing sensor
- Power button to turn the pen on/off

Pattern Recognition

- Selected Algorithm is Dynamic Time Warping
  - Finds optimal matches between time series
  - Used in other pattern matching applications
  - Uses dynamic programming to find ideal paths

Pattern Recognition

Design: Software

Design Calculations

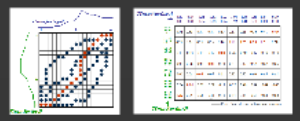
- Data Filtering: 44 FLOPS
- DTW algorithm:
  - time cost: O(n^2)
  - space cost: O(n)
- Battery Runtime on 8 h wake/ 4 h sleep cycles
  - Coin Cell Battery: ~3.00 h
  - Rechargeable AA:~29.40 h
- I/O rates: 36 bits/interval
- Xbee modules: 625 bits per 2.5 ms
- Flash memory required per character: ~3.5 kB
- Onboard Flash: 4 MB ~= 1165 patterns

Challenges Thus Far

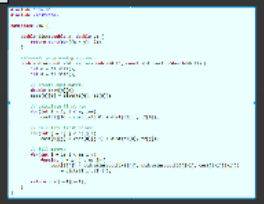
- The Bluetooth module could not act as a Master to poll data from the accelerometer without a microcontroller so we decided to use wireless Xbee module instead.
- Trying to fit all components into a reasonable sized pen body.
- Accurately prediction of accelerometer output data

Design: graphical hierarchy of source code

Design: Software



Awesome Code



Optional Features

- Extra button to change pen functionality to perform basic arithmetic
- Option to allow pen to be user dependent (calibrates pen according to users initial written input pattern)
- Ability to recognize alphabets (in future - handwriting)
- Design a GUI for character display on the computer
- Sensor activated without use of a button trigger or pressure sensor
- Incorporate Bluetooth and integrate this new design so that users can connect the pen to their smartphones and generate outputs to an app

Components

- Implementation on DE2 Board:
  - Xbee transceiver module
  - USB connection to computer
- Implementation on Pen Unit:
  - Xbee transceiver module
  - Accelerometer
  - Power Source
  - Switches (Trigger, Power)

Results of Unit and Integration Testing

DTW algorithm test
- Test input with 400 data points
- runs in 0.172s using an i7@3.4GHz

Application Notes

-Although we do not have application notes of our own yet, we are using an Xbee module from a past project on 2012.

Design: Hardware