

Caitlin Ramsey
Application Note
ECE 480
Design Team 4: Accessible Insulin Pump Design
November 14th

Executive Summary

Capacitive touch can be found in many areas of design. Not only is it helpful, but it makes the user experience more modern and sleek. Integrating capacitive touch into our design will help our design become accepted by users.

Table of Contents

Introduction	3-4
Materials	4-5
Programming	5-7
Circuit Design	7
Conclusion	8
References	9

Introduction

Capacitive touch comes in many forms for different uses. The basic idea behind capacitive touch is there is a system that contains its own capacitance. When an external object with its own capacitance, such as a finger, comes near or touches the system, the capacitance of the system is changed. The system detects the disturbance of capacitance and responds with the appropriate actions given to it by the designer. Below, in Figure 1, is a picture of a basic capacitive touch sensor. It has an insulator with a thin film of conductive coating. The conductive coating is on the opposite end of where the sensor will be touched. Then there is a microcontroller that detects the change in capacitance below the conductive coating.

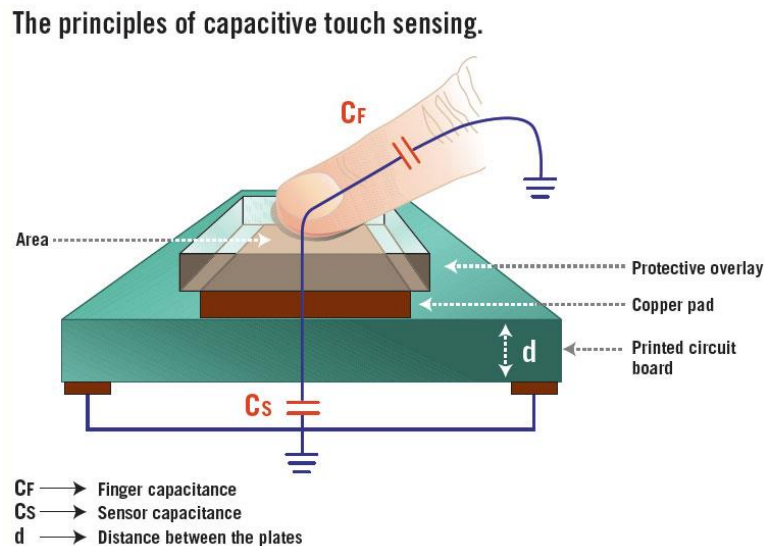


Figure 1

There are multiple types of capacitive touch systems, such as projected capacitance that includes self-capacitance and mutual capacitance and surface capacitance. In the case of the insulin pump, surface capacitance will be used. Surface capacitance has a uniform coating of conductive film on the insulator, as opposed to projected capacitance that has a grid, and uses sensors at the four corners of the screen to detect the location of where the sensor was touched.

Capacitive touch will be used in the insulin pump design in order to help the user understand what each buttons' function is. When the screen changes, the user can lightly touch the button, without pressing it, and hear what the button's function is for the current screen. In order to implement this, conductive tape will be put on each button and the tape will be attached to the microcontroller. The buttons, and thus the tape, will not be connected to one another, so the microcontroller does not need to distinguish between locations on a screen. Consequently, this will be an attenuated version of surface capacitive touch. The conductive film will still be uniform, but the sensor will only need to distinguish between different, isolated buttons instead of locations within a screen. In this application note paper, I will discuss how to make a simple capacitive touch system with an MSP430G2553 Launchpad.

Materials

Microcontroller

As stated in the introduction, a microcontroller will be needed to detect the change in capacitance of the system. In the case of the insulin pump design, I used an MSP430G2553 Launchpad, pictured in Figure 2. The Launchpad contains UART connections, which allow it to communicate with the RC Systems V-Stamp being used in the current design. It also is compatible with Energia software, making it simple and easy to program.

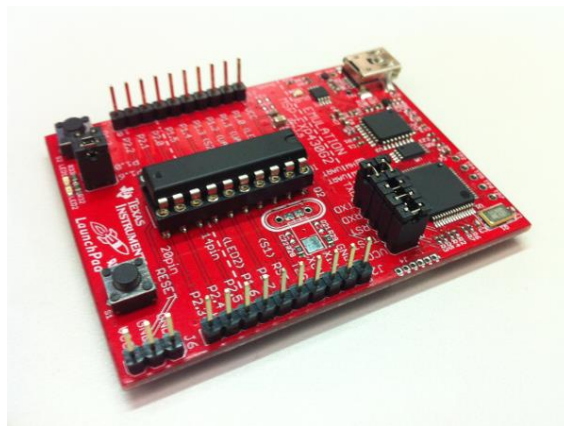


Figure 2

Conductive Film

In order to give the buttons on the insulin pump capacitive characteristics, a conductive film can be put on top of the buttons. In this case, I used conductive tape. The conductive tape will be located on the buttons and wires will be soldered to them, so they can make a connection between the microcontroller and button.

Insulated Coating

In order to keep the buttons from detecting stray capacitance, an insulated coating needs to be placed on top of the conductive film. In this design, stickers should work fine.

Connection

The connection between the microcontroller and the conductive film will be insulated wire.

Programming

Code Composer Studio

Code Composer Studio is a programming language associated with the MSP430. It was not used in this design because of its cumbersome structure. I will be focusing on a different programming language.

Energia

Energia is a simpler and sleeker programming language as compared to Code Composer Studio that is associated with the MSP430, and it will be used in the design. Figure 4 contains an example of code used to program capacitive touch into a microcontroller. The MSP430 has an additional capacitive touch booster pack that can be bought separately. With it comes a software library for capacitive touch sensing, CapTouch.h. The capacitive touch sensing library can be found without buying the booster pack and the MSP430 Launchpad can be configured to have capacitive touch without the booster pack. For this reason, CapTouch.h needs to be included in the program.

```

#include <CapTouch.h>

#define BUTTON4 P1_4
#define BUTTON3 P1_5
#define BUTTON2 P2_0
#define BUTTON1 P2_1
#define LED1 P1_0

uint8_t state = false;

CapTouch testbutton1 = CapTouch(BUTTON1, TOUCH_BUTTON);
CapTouch testbutton2 = CapTouch(BUTTON2, TOUCH_BUTTON);
CapTouch testbutton3 = CapTouch(BUTTON3, TOUCH_BUTTON);
CapTouch testbutton4 = CapTouch(BUTTON4, TOUCH_BUTTON);

boolean buttonState1 = false;
boolean buttonState2 = false;
boolean buttonState3 = false;
boolean buttonState4 = false;

void setup() {
  /* Use the middle LED to indicate touch */
  pinMode(LED1, OUTPUT);
}

void loop() {
  if (testbutton1.isTouched() > 0) {
    if (buttonState1 == false) {
      buttonState1 = true;
      tone(P2_3, 440, 100);
      delay(100);
    }
  } else if (testbutton2.isTouched() > 0) {
    if (buttonState2 == false) {
      buttonState2 = true;
      tone(P2_3, 494, 100);
      delay(100);
    }
  } else if (testbutton3.isTouched() > 0) {
    if (buttonState3 == false) {
      buttonState3 = true;
      tone(P2_3, 523, 100);
      delay(100);
    }
  } else if (testbutton4.isTouched() > 0) {
    if (buttonState4 == false) {
      buttonState4 = true;
      tone(P2_3, 554, 100);
      delay(100);
    }
  } else {
    buttonState1 = false;
    buttonState2 = false;
    buttonState3 = false;
    buttonState4 = false;
  }
}

```

Figure 4

The capacitive touch library is based on the idea the capacitive touch sensors are their own buttons. So, the pins that will be used as capacitive touch sensors are labeled as buttons in the program. In this program, the microcontroller will show when touch has occurred through the LED built into the Launchpad. Inside the void (loop) portion of the program, which the microcontroller continually reads, the microcontroller is continually reading to see if the pins have been 'switched'. Using the capacitive touch library, the microcontroller is able to read the capacitance and, with a predetermined capacitance, determine if the 'button' was touched. In this program, the microcontroller emits a tone depending on which button was pressed, but, due to the ease of programming with Energia, the microcontroller can do any action necessary.

Circuit Design

The circuit is relatively simple for a tone generator. Some type of button must be attached to the microcontroller's pins. In my design, I used a breadboard to attach insulated wire to the microcontroller. With the insulation stripped off of the end of the wires, the ends were used as the buttons in my design. Then a speaker must be attached to the output pin, designated as pin 2.3 in code in Figure 4, and ground of the Launchpad. Figure 5 contains a block diagram of the circuit.

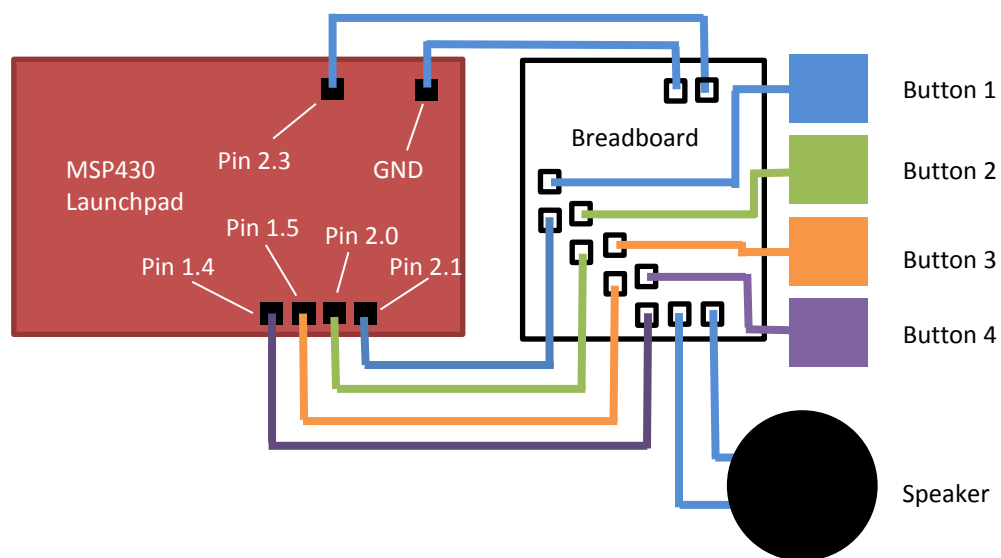


Figure 5

Conclusion

Using the capacitive touch library associated with the MSP430 booster, users can manipulate the MSP430 Launchpad to distinguish between different pins. The circuit and program created in this application note were relatively simple. The main function of the program was to have the microcontroller continually check to see if the capacitance changed on any pin and perform an action based on that pin. The circuit requires a speaker and a conductive material. The simplicity of the program and circuit reflect the flexibility of the design. It can only be used to distinguish between different, isolated positions. This is unlike normal uses for capacitive touch, in which the sensors distinguish between a positions on the sensor relative to another.

References

<http://energia.nu/Serial.html>

<http://forum.43oh.com/topic/3158-energia-library-capacitive-touch-library/>

<http://www.embedded.com/design/mcus-processors-and-socs/4218309/Waking-up-a-capacitive-touch-sensing-device-with-an-MCU-peripheral>