BilkeyJ Week 4 Lab Modified - Fever Detector

Sunday, August 23, 2020 10:19 AM

This uses the LM 34 and MSP 432 to check for fever temperatures. Accurate to 1 degree F.

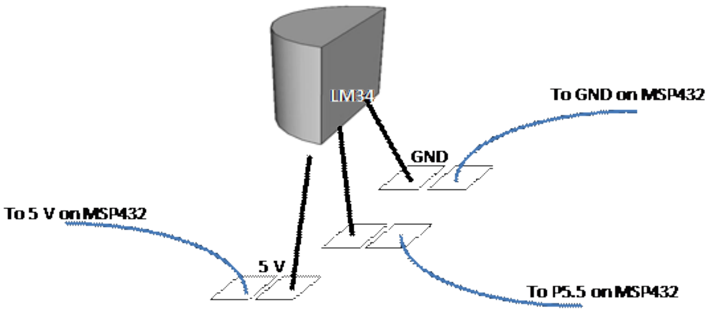
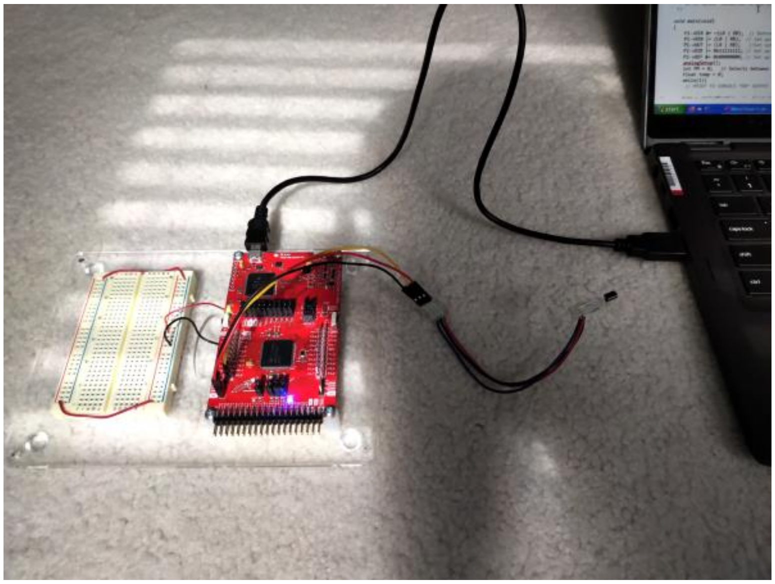


Diagram by Dr. Ross

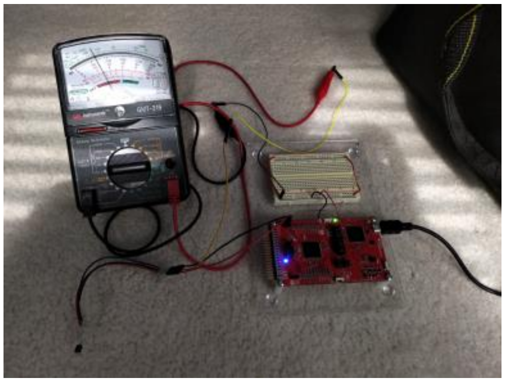


How to use:

1. Connect LM 34 to MSP 432 board as shown in the diagram above. Use jumper wires to give yourself room to move the temperaturesensor around.
2. Install Code Composer Studio on your PC, import the CCS project, build and click debug, then click the green arrow / "play" icon.
3. Insert the LM 34 in your armpit. Wait until the LED turns green, then wait another 20 seconds. If it turns red, your temperature is too high.
4. You no longer need your PC to test your temperature. Connecting the MSP 432 to any power source will run this program. If youconnect the MSP 432 to your computer and have Code Composer Studio running, you can hit the right button on the MSP 432 to output the exact temperature in the console. The LED will turn yellow in this mode.

Notes: The LM 34 sensor is only accurate to 1 degree Fahrenheit, and this project cannot replace a medical thermometer. I still think it is useful for EE students at the Milwaukee School of Engineering who have all these parts already or can easily get them.

You do not need a MSP 432 board to interface with the LM 34 at all. Simply connecting 5v power and the middle pin to a voltmeter will give you a temperature reading. The LM 34 outputs 10mV / degree F.



Using the LM 34 with
a voltmeter

```
#include "msp.h"
#include "ee1910delay.h" // It uses ee1910delay.h to delay execution of further code
#include "ee1910analog.h" // The program uses ee1910analog.h for analog to digital conversions - input and output
#include <stdio.h> // Necessary for non-janky printf

#define LB 0x02
#define RB 0x10

/*
This program will read the output of a LM34 temperature sensor and accordingly light up an LED or print the temperature to co nsole.
The LED color will be blue for a temperature below normal body temp, green for normal body temp, and red for a fever.
By John Bilkey, modified from Lab 4 from Dr. Ross's EE 1910 Class.

  Required Hardware:
  * MSP 432 Board
  * Breadboard
  * LM 34 Sensor connected to pin 5.5
 */

void main(void)
{
    P1->DIR &= ~(LB | RB);  // Buttons are use Port 1 Pin 1 (Left) and Port 1 Pin 4 (Right).  The direction of a button pin is input (0).
    P1->REN |= (LB | RB);  // Set pull-up resistor for button
    P1->OUT |= (LB | RB);   //Set out register for 1 (input)
    P2->DIR |= 0b11111111; // Set up LEDs
    P2->OUT &= 0b00000000; // Set up LEDs
    analogSetup();
    int PM = 0;   // Selects between printing to console (1) or LED output (0)
    float temp = 0;
    while(1){
     // PRINT TO CONSOLE *OR* OUTPUT TO LED BASED ON IF STATEMENTS

    temp = analogRead();  //  float temp is set to analogRead - making sure decimal calculations don't have issues
    temp = ((temp*3.3)/(40.95));  // Convert analogRead output to degrees Fahrenheit

        // CONTINUE IF PRINT TRUE
    if (PM == 1){
        analogWrite(0,50); // R
        analogWrite(1,50); // G
        analogWrite(2,0); // B
            printf("Degrees F: %f\n",temp); // Print conversion output to console
            delay(500); // Wait 500ms
    }

   // CONTINUE IF PRINT FALSE
   if (PM == 0){

        //BEGIN LED OUTPUT
        if (temp<97){
             analogWrite(0,0); // R
             analogWrite(1,0); // G
             analogWrite(2,100); // B  Colder than Body
        }

        if ((temp>97) & (temp<99)){
             analogWrite(0,0);
             analogWrite(1,100);
             analogWrite(2,0); // Normal Body Temp
            }

        if (temp>99){ // 99 Degrees F is the limit I set, you can change this.
             analogWrite(0,100);
```

```
           analogWrite(1,0);
           analogWrite(2,0); // Fever
      }
  }

      // EXECUTE REGARDLESS OF PM VALUE
      delay(500); // 500 ms delay

      // RIGHT BUTTON - Sets PM to 1 or 0.
      // 1 = Print Temp to Console 0 = LED Color Output (Default)
      if ( (P1->IN & RB) == 0) {
          if (PM == 1) PM=0;
          else {
              analogWrite(0,0);
              analogWrite(1,0);
              analogWrite(2,0);
              PM=1;
          }
          while((P1->IN & RB) == 0){} // wait until button is not pressed
      }
 }
}
```