Author: Wen Niu

Application Note

Introduction

My job was responsible for soldering the PCB and cooperate with my teammates in programming the signal process. Also, I worked on the data collection for the radar system.

Our team used the 32 bit ARM processor Teensy 3.1 to processing the signal. Also, we used the teensy to generate the triangle wave. Finally, we will display the result on a 16 * 2 LCD monitor.

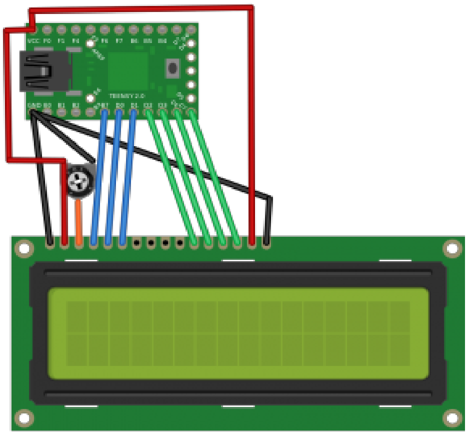
Teensy 3.1 introduction

Teensy 3.1 does not only process faster than the Arduino, but also Teensy 3.1 board has a useful library named FreqMeasure. This library works efficiently for measuring the frequency from 0.1 Hz to 20 kHz. Furthermore, we uses the LiquidCrystal Fast library to display the result on the LCD monitor.
LiquidCrystalFast library lets us use small character type displays. Characters and a limited set of custom symbols can be used.

Code Analysis

Basic Command meaning:
LiquidCrystalFast lcd(RS, Enable, D4, D5, D6, D7) -- Create the LiquidCrystal object and specify the seven pins where the LCD is connected.
lcd.begin(width, height) -- Initialize the display and set the size.
lcd.print(anything) -- Print a number or text. This works the same as Serial.print(), but prints to the LCD.
lcd.setCursor(x, y) -- Move the cursor to position (x, y). These are zero-based coordinates.

The method of connection



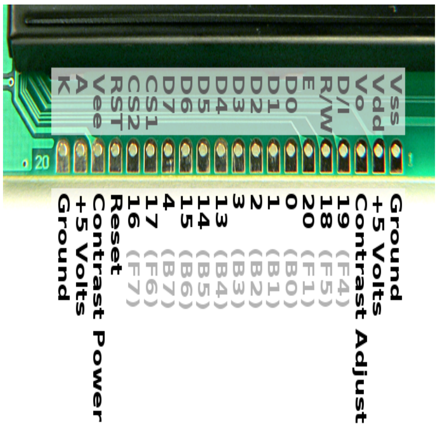
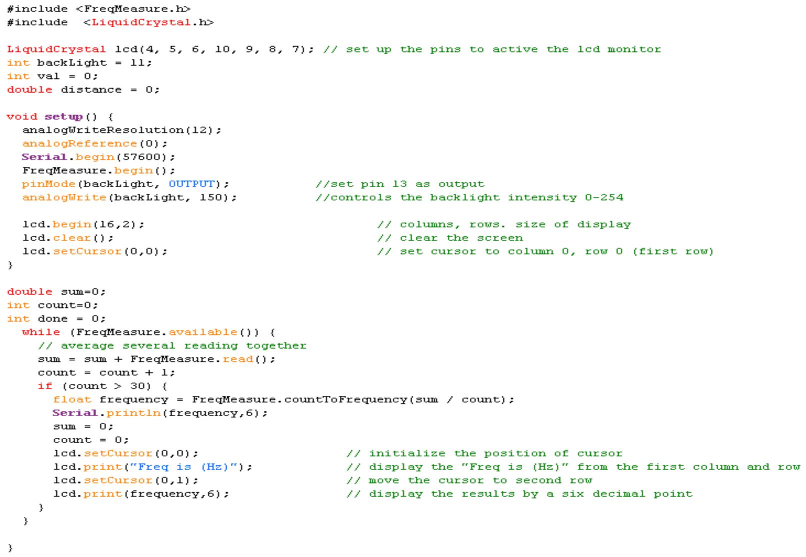


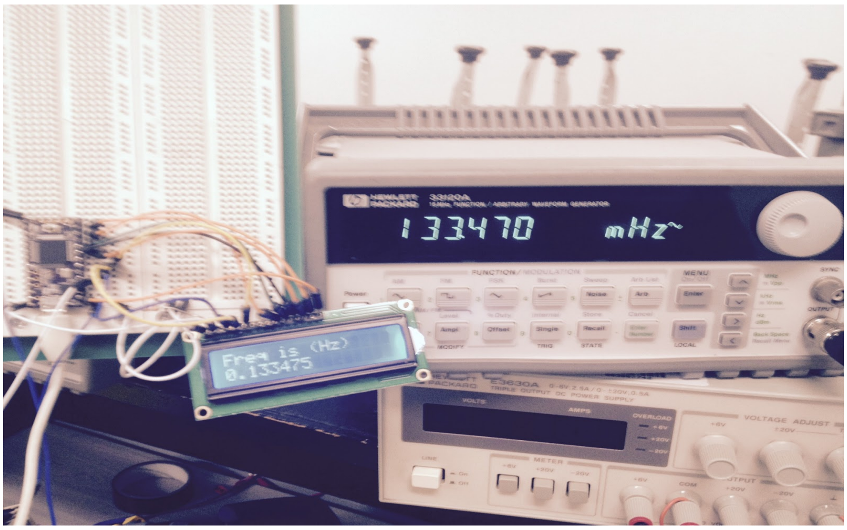
Figure 2 is the explanation of the code:



```
#include <FreqMeasure.h>
#include  <LiquidCrystal.h>

LiquidCrystal lcd(4, 5, 6, 10, 9, 8, 7); // set up the pins to active the lcd monitor
int backLight = 11;
int val = 0;
double distance = 0;

void setup() {
  analogWriteResolution(12);
  analogReference(0);
  Serial.begin(57600);
  FreqMeasure.begin();
  pinMode(backLight, OUTPUT);           //set pin 13 as output
  analogWrite(backLight, 150);          //controls the backlight intensity 0-254

  lcd.begin(16,2);                              // columns, rows. size of display
  lcd.clear();                                  // clear the screen
  lcd.setCursor(0,0);                           // set cursor to column 0, row 0 (first row)
}

double sum=0;
int count=0;
int done = 0;
  while (FreqMeasure.available()) {
    // average several reading together
    sum = sum + FreqMeasure.read();
    count = count + 1;
    if (count > 30) {
      float frequency = FreqMeasure.countToFrequency(sum / count);
      Serial.println(frequency,6);
      sum = 0;
      count = 0;
      lcd.setCursor(0,0);                   // initialize the position of cursor
      lcd.print("Freq is (Hz)");            // display the "Freq is (Hz)" from the first column and row
      lcd.setCursor(0,1);                   // move the cursor to second row
      lcd.print(frequency,6);               // display the results by a six decimal point
    }
  }

}
```

Figure 2

The result demonstrates below.



Display using fast 7 signal connection

Data Collection Analysis

The datas and analyze have been uploaded to group report. For the data collection, the most difficult problem is connection. Because the sizes of components and our PCBs are too small, we spend a lot of time to debug the connections.