Juan Gonzalez Aguayo & Kun-Pu Lee
Oscilloscope ECG Report

Description

In this project, we modified our existing ECG program to embed the MAX11300 PIXI and use its DAC to display the cardiac cycles on an oscilloscope. The adjustments were made by deleting the code that enabled us to display 10 minutes data as we only needed the continuous ECG lead voltage readings. The MAX30003 reads the small cardiac voltage and converts it into a binary number which we send to the Pixi's DAC. Because the MAX30003 binary readings contained negative numbers on a large y-axis (-7000 to 7000) and the Pixi's DAC required data from 0x000 to 0xFFF (0 to 4095) we wrote a function to map each ECG binary number to the corresponding DAC input value.

System requirements

1. Display the continuous cardiac cycles on an oscilloscope using the MAX11300 PIXI to interface the MAX30003 and MAX32630FTHR.

System Diagram

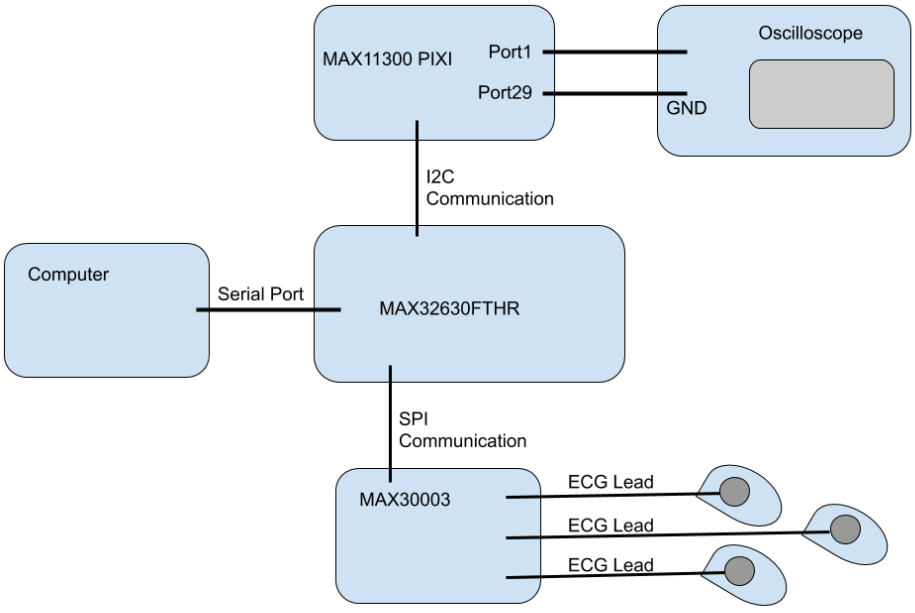

Figure 1: ECG Diagram

System Software Flow Chart

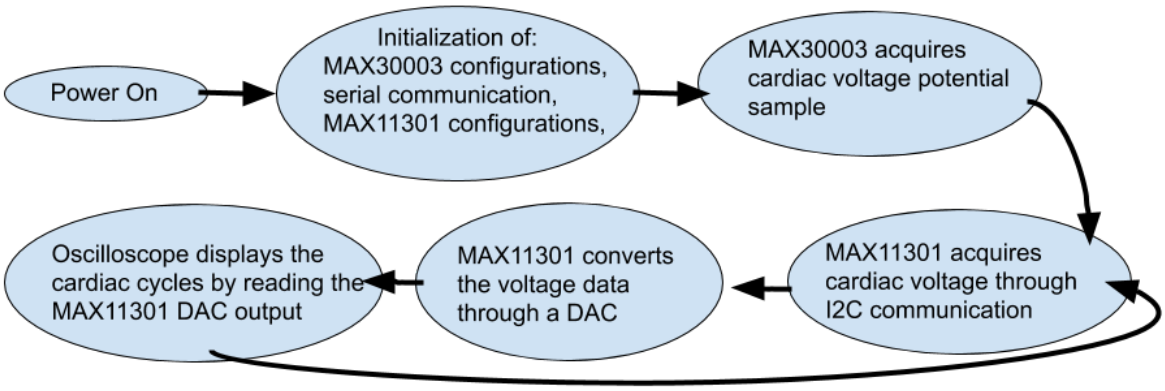

Figure 2: ECG Software Logic and Flow Chart

System Performance

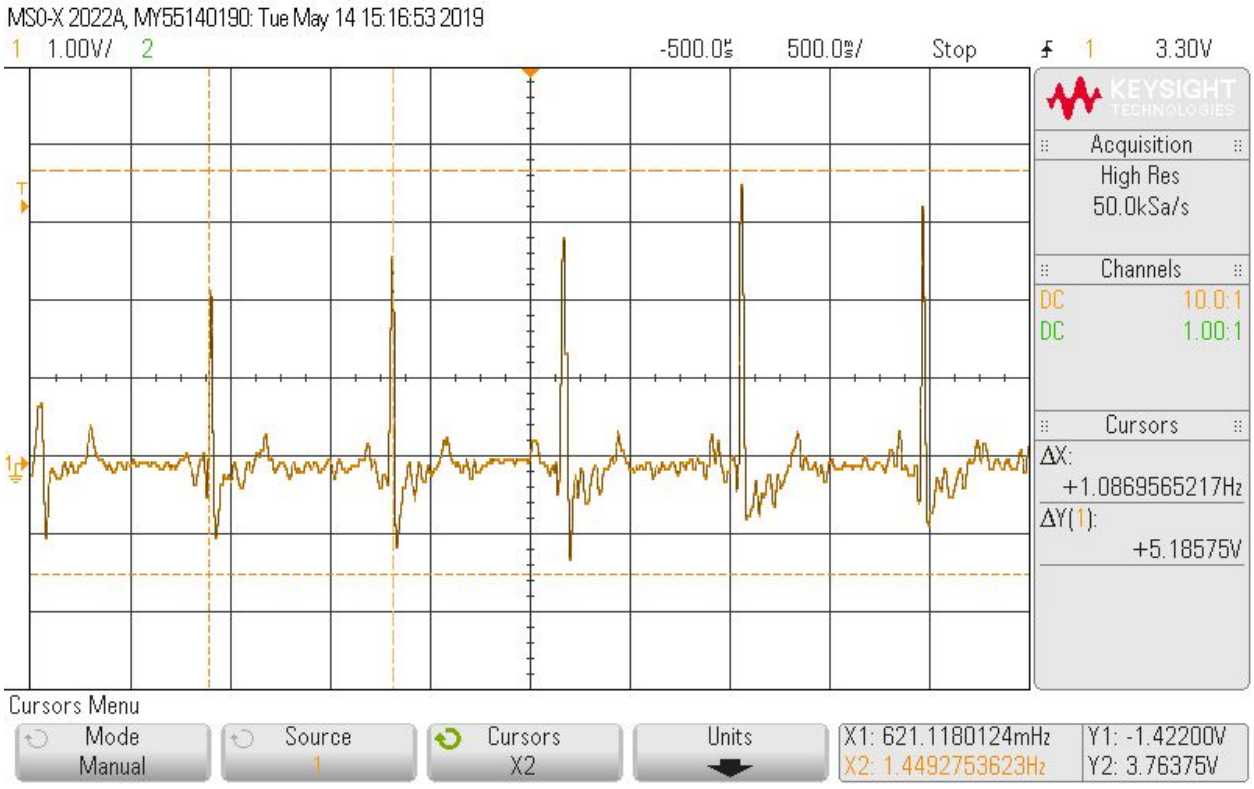

Figure 3: Oscilloscope Capture of the ~1Hz Cardiac Cycles

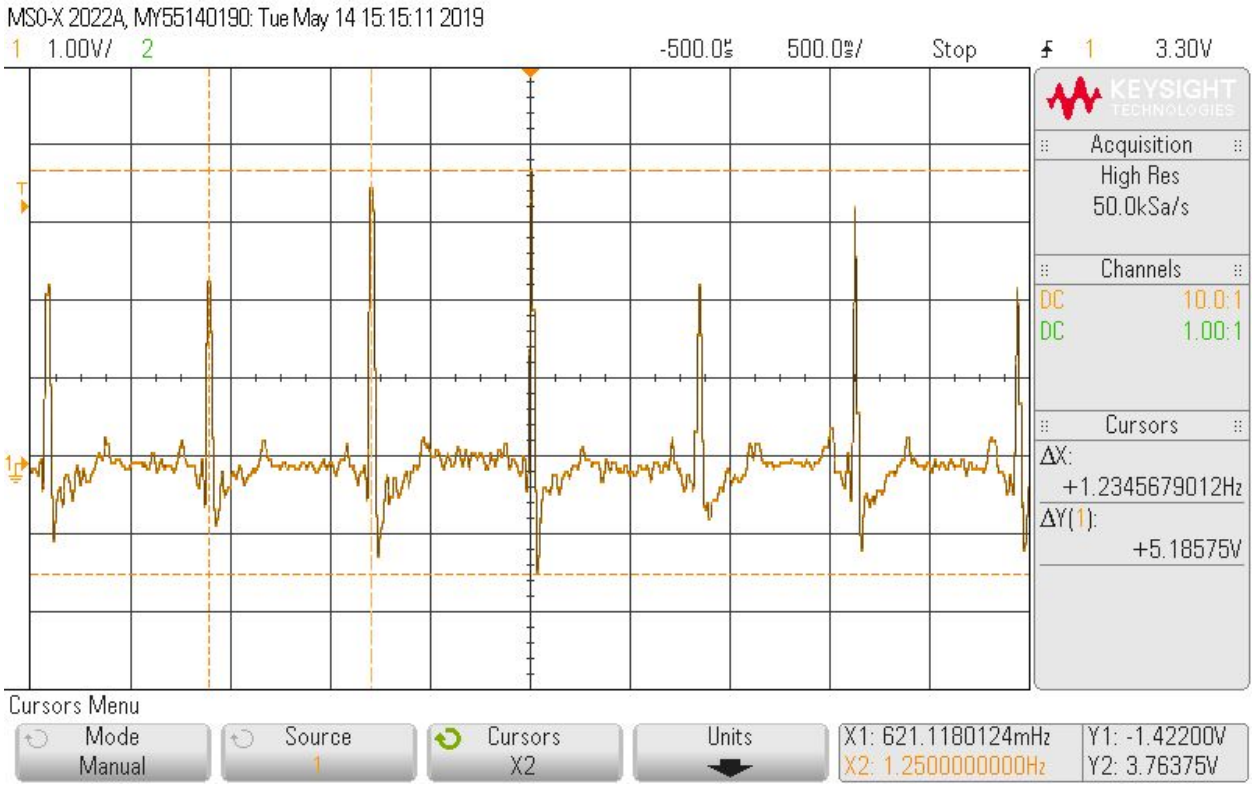

Figure 4: Oscilloscope Capture of the electrocardiogram

Code Snippet

The following code obtained from source [2] initiates and interfaces the I2C communication between the MAX11301 and the MAX32630. The DAC on MAX11301 has configured to output a voltage between 5V and -5V.

```
I2C i2cBus(I2C1_SDA, I2C1_SCL); // I2C bus, P3_4 = SDA, P3_5 = SCL
MAX113XX_I2C pixi(i2cBus, MAX113XX_I2C::MAX11301, 0x38, P5_5);
pixi.dacWrite(MAX113XX_Pixi::PORT0, 0x000);     // Pixi PORT0 is -5V
pixi.dacWrite(MAX113XX_Pixi::PORT1, 0xFFF);     // Pixi PORT1 is +5V
```

The following is a modification of the MAX30003 ECG's example [4]. We sent the cardiac voltage sample to the MAX11301 DAC for processing as it became available. A delay was added to allow the MAX11301 to process the current cardiac voltage sample data. In other words, we need to wait before a new sample's voltage data is introduced to the MAX11301. On the other hand, the absence of the delay produced a corrupted oscilloscope reading of the cardiac cycles.

```
for( idx = 0; idx < myData.readECGSamples; idx++ ) {
        wait(.007);
```

```
pixi.dacWrite(MAX113XX_Pixi::PORT1,int((myData.ecgSample[idx]+7000)*4095/14000)
);
```

The following code maps the y-axis-data-points from the MAX30003 ECG to the y-axis-voltage-values that the MAX11301 DAC is able to supply (-5V to 5V).

```
int( (myData.ecgSample[idx] + 7000)*4095/14000 )
```

Because the ECG data points contained negative numbers, we took the y-axis data and added 7000 to shift all data points in the positive y-axis. This was necessary since the Pixi does not allow for negative hex number representation; as it maps from 0x000 ~ -5V to 0xFFF ~ 5V. We divided this number over 14000 to "shrink" the data into units and multiplied this by 4095 to map each individual unit to the new y-axis (4095 = 0xFFF = 5V).

$$\text{Y-axis-voltage-Value} = \frac{(MAX30003\ ECG\ Reading + 7000) \times 4095}{14000}$$

Conclusion

One of the biggest problems we encountered was accidentally shorting the Pixi by touching to of its leads with the oscilloscope probe. When shorting occurred, the board stopped working and a red LED indicator staying on whenever the board was plugged into a power source. After navigating the Pixi data sheet [5] we concluded that the board incorporates smart protection against this type of occurrence. The seriousness of the shorting determines if the board can configure automatically after an indicated time and reset itself. Because our short was not permanent, all we had to do was wait one day for the board to function again.

Another problem we encountered was when sending the voltage data to the pixi's DAC. Initially, we sent the data to the Pixi as it became available. However, it produces a corrupted oscilloscope capture. We concluded that this happened because we did not allow the Pixi's DAC to process the present voltage data before sending new data. After including a delay, the oscilloscope displayed the data as we expected. To visualize and double-check our oscilloscope readings, we also ran the ECG continuous plot from a computer. Both plots matched and seemed coherent. Since we were using a DAC, if one zooms into the oscilloscope then one can see the voltage steps from the DAC.

Reference

[1] MAX11301WING Data Sheet
https://datasheets.maximintegrated.com/en/ds/MAX11301WING.pdf

[2] MAX11301WING Demo Example
https://os.mbed.com/teams/Maxim-Integrated/code/MAX11301WING_Demo/

[3] MAX30003WING Data Sheet
https://datasheets.maximintegrated.com/en/ds/MAX30003WING.pdf

[4] MAX30003 ECG Example
https://os.mbed.com/teams/MAXIM-TTS-Code/code/MAX30003_Demo_Debug/shortlog/

[5] MAX11301 PIXI, 20-Port Programmable Mixed-Signal I/O with 12-Bit ADC, 12-Bit DAC, Analog Switches, and GPIO
https://datasheets.maximintegrated.com/en/ds/MAX11301.pdf