Davis 109006201
周志偉

Homework 1 Digital and DAC Interface Report

1. Please write a mbed program to show Morse codes with a LED

- Instead of using any loop, I decide to open and close each led manually until the word is complete. I decided to write the character " HI SOS". After each character in the same word will have a 3 sec pause, but if the word changes it will have a 7 sec pause. In the same character the normal pause will be 1 sec. I found no problem with this assignment.

2. Please write a mbed program to show Morse codes with a LED with button

- It took me a long time to think of a way to implement this. At first I wanted to have a variable to count the amount of each button press and output the led, but then I have the problem of differentiating between the short button and the first press of my long button. Because of this I decided to use a timer that loop forever and for every 4 sec it will print start pressing. After each round the flag(variable to count the number of button presses) will be sent to the output. To stop the long button from lighting when the first button is pressed, it sets the output so that it will light after each round instead of lighting immediately after pressing. The flag will be reset after each round too.

3. Please write a mbed program to show Morse codes with a LED, 7 segment display

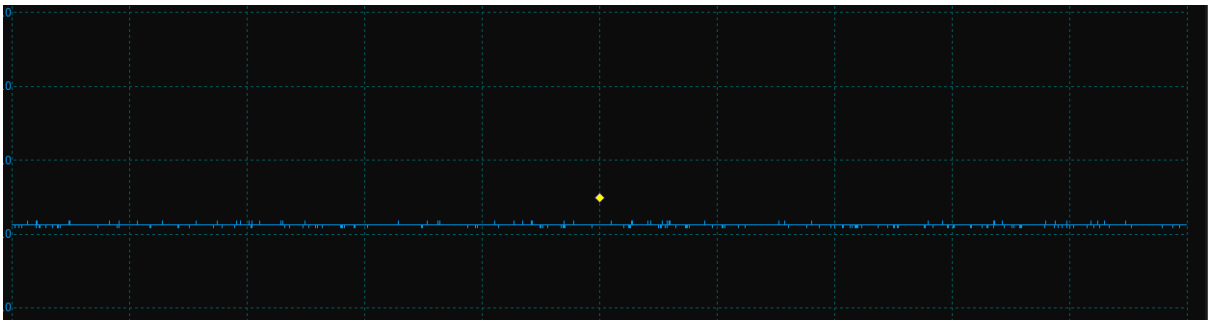
- We use the same code from above, so I decided to use the word "SOS" for this assignment. After 3 short presses the letter "S" will be displayed and after 3 long presses "O" will appear. If the long button was pressed after the short press button, it will reset and start counting again. Same for the short press button, if a long button is pressed it will be reset.

4. rewrite the sawtooth waveform generator with write_u16(unsigned short)

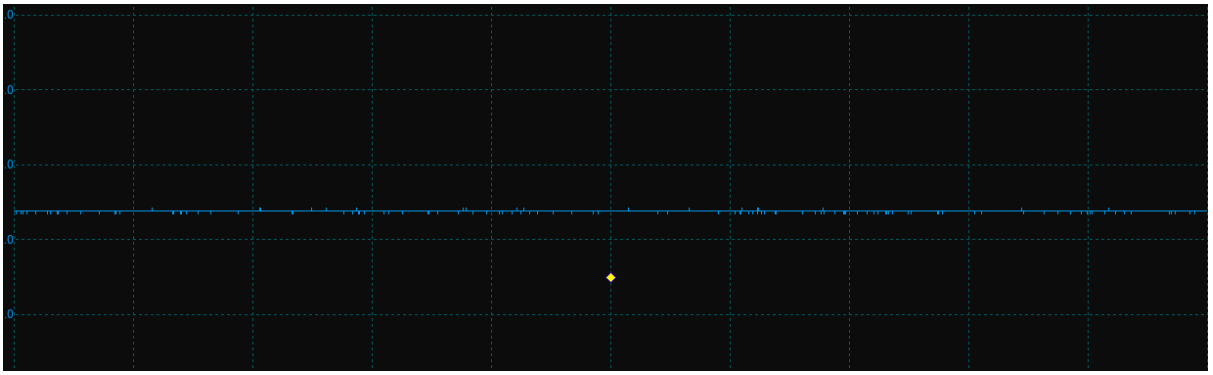
- In the original sawtooth waveform it will change the voltage on the digital output pin by 0.1 * VCC and print what the measured voltage should be (assuming VCC = 3.3v). It will then turn on the led if the voltage is greater than 0.5f * VCC. The thread will then sleep for 1 second.
- I change the range in the for loop from 0 to 0x0000 and 1 to 0xFFFF, and in order to use write_u16 we have to change the variable from float to uint16_t. I increment each loop by 0x1000. The output will then print 16 output starting from 0, and the LED won't be changed.

```
aout = 0.000000 volts
aout = 0.206300 volts
aout = 0.412601 volts
aout = 0.618901 volts
aout = 0.825202 volts
aout = 1.031502 volts
aout = 1.237802 volts
aout = 1.444103 volts
aout = 1.650403 volts
aout = 1.856703 volts
aout = 2.063004 volts
aout = 2.269304 volts
aout = 2.475605 volts
aout = 2.681905 volts
aout = 2.888205 volts
aout = 3.094506 volts
```

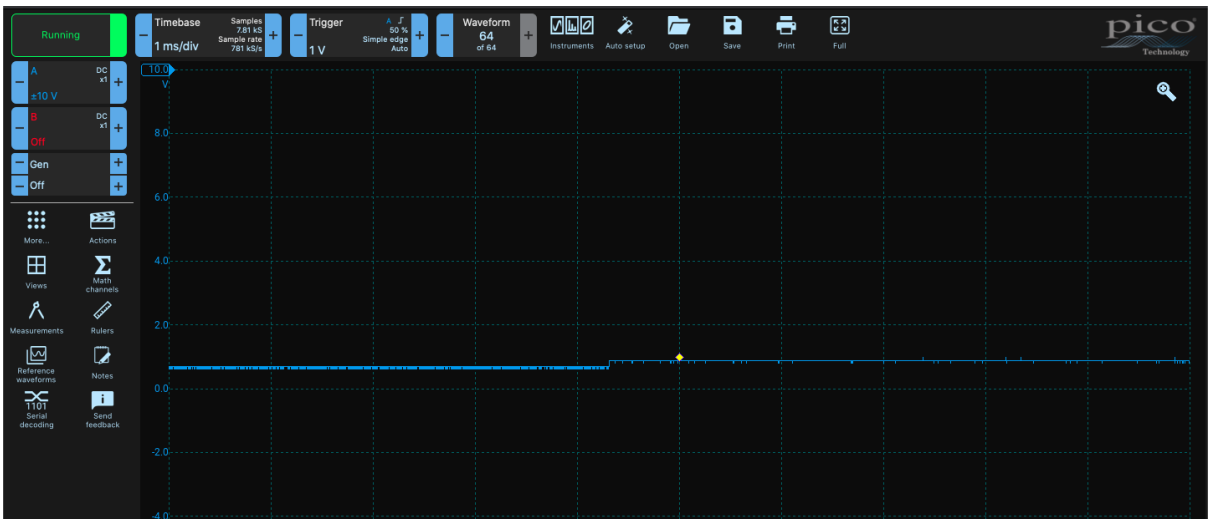
When picoscope is zero volts



When it at 3.09 volts



5. Please use wait_us() instead of "ThisThread::sleep_for()"
- I just change ThisThread::sleep_for(1s) to wait_us(1), the output will print at a faster rate.


Running
Timebase
1 ms/div
Samples
7.81 kS
Sample rate
781 kS/s
Trigger
1 V
50 %
Simple edge
Auto
Waveform
64
of 64
Instruments
Auto setup
Open
Save
Print
Full
pico
Technology
A
DC
x1
±10 V
B
DC
x1
Off
Gen
Off
More...
Actions
Views
Math channels
Measurements
Rulers
Reference waveforms
Notes
Serial decoding
Send feedback
10.0
V
8.0
6.0
4.0
2.0
0.0
-2.0
-4.0