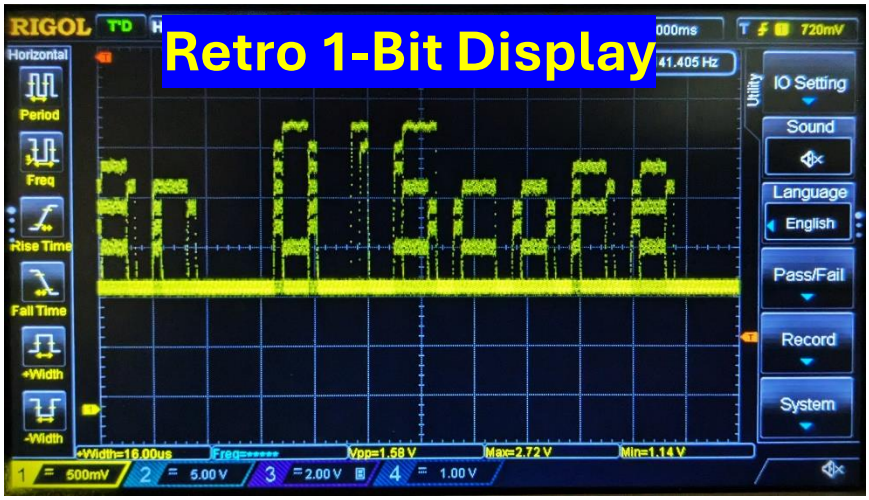
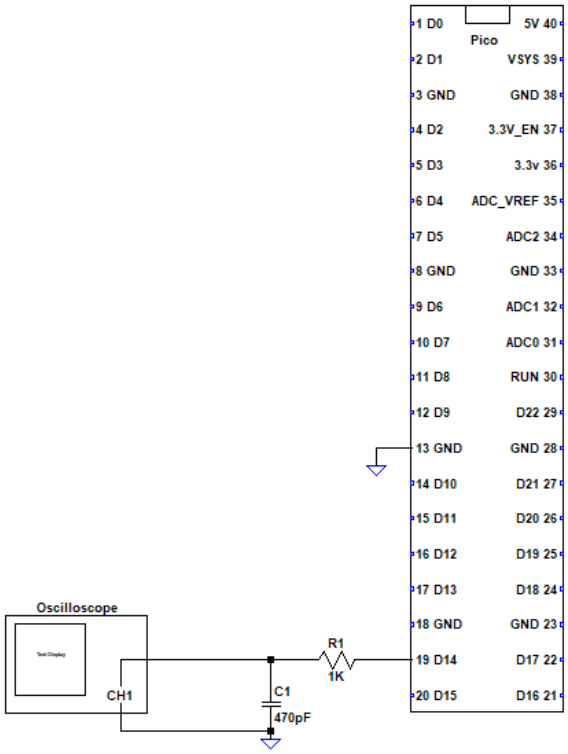


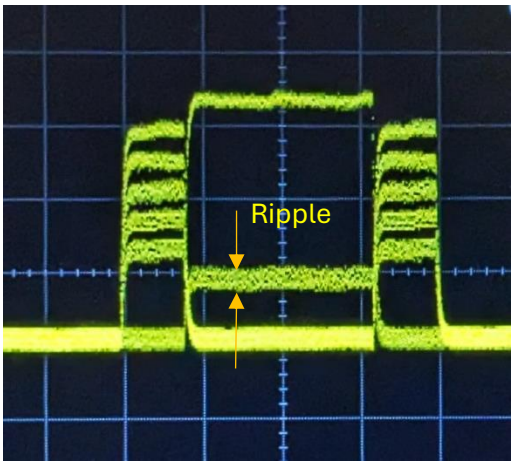
Make an O'scope text display using only one Pico I/O pin. You will need a Raspberry Pi Pico with pins, a USB cable, a 1Kohm resistor, a 470pF capacitor and a couple jumper wires. This is a breadboard project. Use any size that you have.



The display is a seven-line raster scan. To eliminate intensity modulation, the ON pixel is stepped to the scan line from a base level then returned to the base for the OFF level. Seven sweeps are required to form the text line. Analog scopes have a persistence in the phosphor that retain the line scans. They are ideal for viewing. Digital scopes may require some set up to form a good image.

Each line start is triggered on a rising edge from zero volts. This is followed by a front porch delay. Next pixel data is sent for 50 characters. A back porch delay is inserted. The voltage then drops back to zero.

The Pico processor does not have a digital to analog converter. Instead, we use a very fast pulse width modulated (PWM) output running around 8 Megahertz. The PWM has 16 levels or four bits of resolution. The 3.3V PWM signal is reduced to a DC level using an RC lowpass filter. The 1K resistor and 470pF capacitor filter PWM ripple to around 140mVpp. The bit changes will still be distinct as each bit step is 206mV. The partially filtered PWM allows a fast rise and fall time.



The Pico code is written using the Arduino IDE and Earl Philhower's Arduino-Pico. The processor clock is set to 150MHz, the optimization to -O and the USB Stack to Pico SDK. Several Pico libraries are required.

```
#include "pico/stdlib.h"
#include "hardware/pwm.h"
#include "hardware/clocks.h"
```

We want the PWM slice to be available after setup and we use a global variable.

```
uint slice_num ;
```

Set up code performs the following:

- Pin 14 is set to the PWM output.
- The PWM wrap is set to 15 which gives 16 levels.
- The clock divider is set to one to give a PWM clock of around 8MHz.
- The PWM value is set to zero then enabled.

```
gpio_set_function(14, GPIO_FUNC_PWM);
slice_num = pwm_gpio_to_slice_num(14);
pwm_set_wrap(slice_num, 15);
pwm_set_clkdiv(slice_num,1);
pwm_set_chan_level(slice_num, PWM_CHAN_A, 0);
pwm_set_enabled(slice_num, true);
```

The Core-2 program generates the raster scan by writing synchronization and pixel levels to PWM pin-14. Pixels are locked to the microsecond system clock by testing the counter value bits. The delay is twice the SYNCBIT value. SYNCBIT must be a multiple of two.

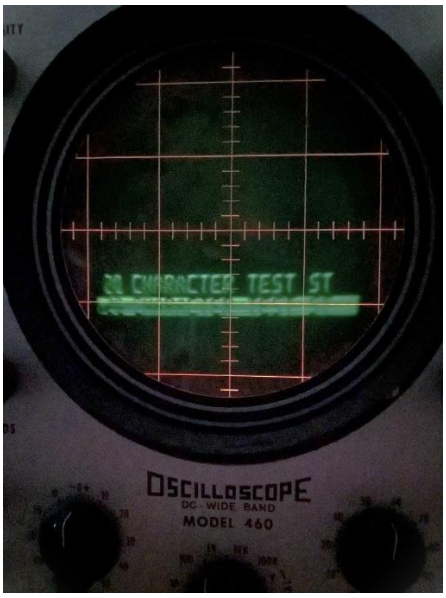
```
pwm_set_chan_level(slice_num, PWM_CHAN_A, x);

while ((time_us_32() & SYNCBIT)!= 0);
while ((time_us_32() & SYNCBIT) == 0);
```

Character formation uses a 5x7 matrix lookup table. Most of the program code is the 128 ascii character values. Characters are five pixels wide with two blank pixels for spacing. Pixel transitions are synchronized to the system microsecond clock register at 16us. For delay times other than 2^N, the microsecond delay function may be used.

```
//A
bc[ 0x41] [ 6 ] = B00100;
bc[ 0x41] [5 ] =  B01010;
bc[ 0x41] [4 ] =  B10001;
bc[ 0x41] [3 ] =  B11111;
bc[ 0x41] [ 2 ] = B10001;
bc[ 0x41] [ 1 ] = B10001;
bc[ 0x41] [ 0 ] = B10001;
```

Since the display generator runs on core-2 this leaves core-1 for processing other tasks. In our case, core-1 is used to receive serial data and store characters in the ascii line array.

Display on an analog O'scope may require a reduced character count. The bandwidth and trigger limitations of the Eico Mode 460 oscilloscope limit the display to 20 characters.



Digital oscilloscopes may not have a variable horizontal time scale. To display on a fixed horizontal scale, the program delays need adjustment where the number of characters plus sync pulse will exceed the horizontal time per division multiplied by the number of divisions. The time displaying characters must be slightly less.

Character Time = CT = 16us * number of characters * 7 + front and back porch delays
Scan Time = ST = number of divisions * time per division
Retrace Time = RT = Sync pulse low time
Line Scan = LS = CT + RT

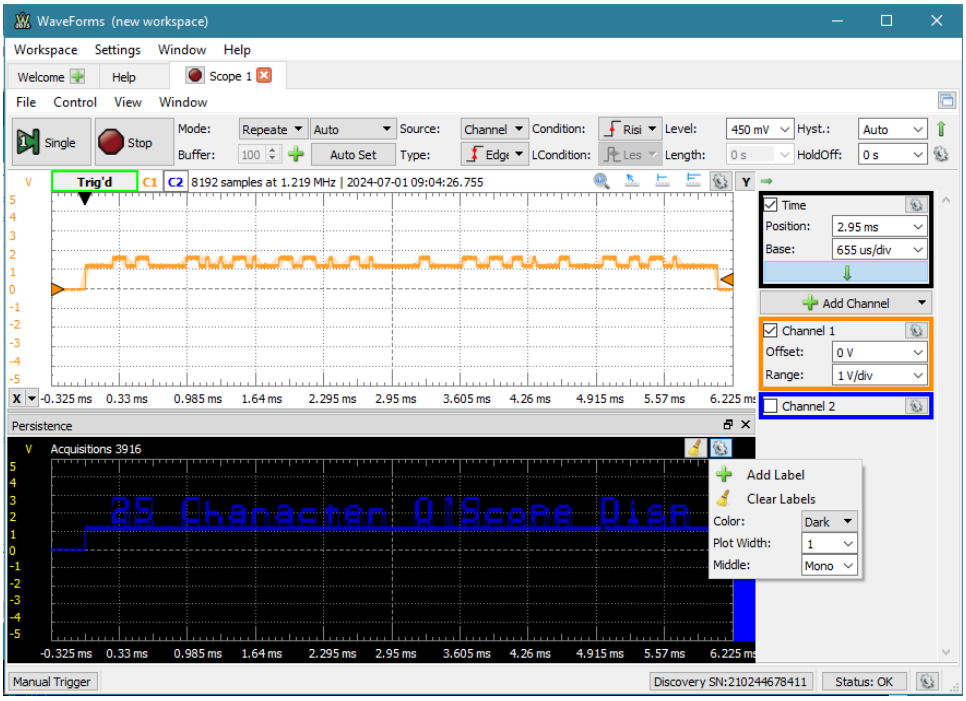
CT < ST < LS

For our Rigol DS1074 O'scope, we have a 1000us sync, two 240us delays and 50 character times 7 times 16us per pixel. This results in a Line Scan period of 7.08ms. The O'scope was set to 500us per division. With 12 divisions, the Scan Time is 6.00ms. The Character Time is 5.6ms.

In addition to triggering on the line scans, display options may need setting. Large memory storage depths cause display delays and rolling dark bands. Low values can fix this problem.

Dot mode displays look better than Vector mode as the rising and falling edges do not overlap the character display.

Some digital O'scopes may not have a Dot display mode. A Persistence display was used with this Digilent® Analog Discovery™ O'scope. To obtain a clearer image, the color mode was set to Dark with a Mono trace color. The number of characters was reduced to half at 25.



Source code and the program file scopewriter_pico50pwm.uf2 are located at:

https://github.com/simple-circuit/retro