

This document shows technical details of the Seuss display prototype. Our design goals:

- 1) Inexpensive off-the-shelf components
- 2) Provide an integration to run animated and musical shows
- 3) Publish the design under an open-source license so all Makers may benefit from it

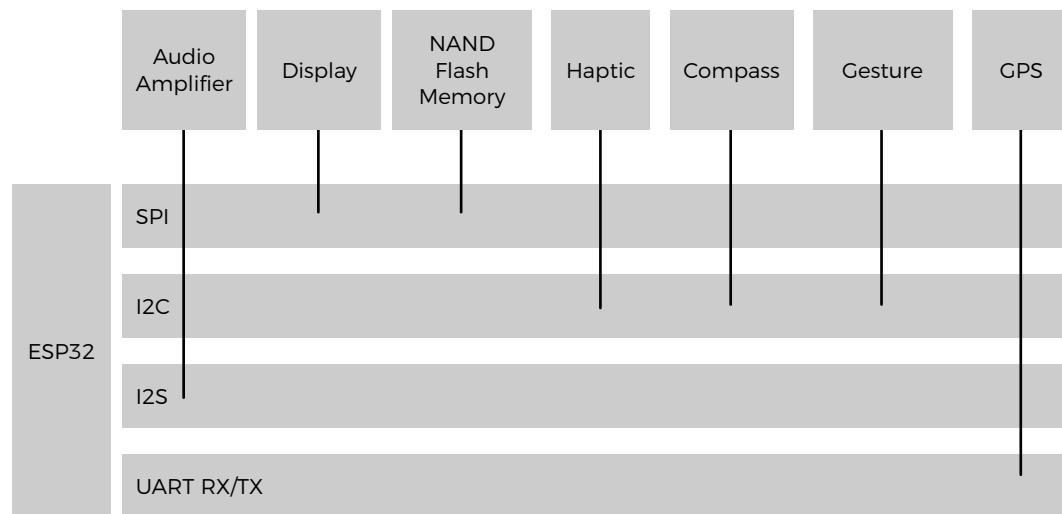
Sox has: ESP32, GC9A01 1.28 Inch Round LCD display, APDS9960 gesture sensor, BNO055 IMU, max98357 audio amp, DRV2605 haptic motor driver, 512 Mbyte SPI Flash memory (NAND, <https://www.adafruit.com/product/4899>), and GPS (TELO132)

Software for Sox is at <https://github.com/frankcohen/ReflectionsOS/tree/main/Devices/Sox/Software>

Sox is a general purpose ESP-32 main board to power storytelling experiences. I named the board versions from Dr. Seuss' writing because Theodor Seuss Geisel was turned down 28 times before being published. I've tried many times to build the display for the Reflections project.

None of this is complicated. However, putting it all together was difficult and time consuming. For example, all but one GPIO pin is used in this design.

The design uses mostly breakout boards from Adafruit. They provide needed pull-up resistors and level-shifters. The finished board is entirely 3.3 volts and delivered in a 36 mm round board.





<http://www.hiletgo.com/ProductDetail/1906566.html>

ESP-WROOM-32 ESP32 ESP-32S Development Board 2.4GHz Dual-Mode WiFi + Bluetooth Dual Cores Microcontroller

The ESP32 integrated with Antenna switches, RF Balun, power amplifiers, low-noise amplifiers, filters, and management modules, and the entire solution occupies the least area of PCB. 2.4 GHz Wi-Fi plus Bluetooth dual-mode chip, with TSMC Ultra-low power consumption 40nm technology, power dissipation performance and RF performance is the best, safe and reliable, easy to extend to a variety of applications.

Espressif - based in Shanghai, China - makes ESP32. It's an open-source design.

Software drivers and utilities for ESP32 are at:

<https://github.com/espressif/arduino-esp32>

<https://github.com/nodemcu/nodemcu-firmware/tree/dev-esp32>

Notes on using ESP32 with Arduino IDE:

Follow these instructions to install ESP32 to Arduino IDE

<https://github.com/espressif/arduino-esp32/blob/master/docs/arduino-ide/mac.md>

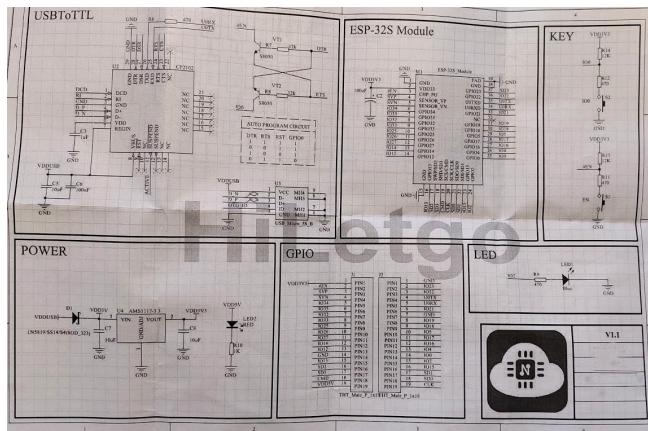
Use ESP32 Dev Board for Board selection

To reference GPIO pins in code use just the number, for example "digitalWrite(13, HIGH)" sets GPIO13 high.

~/Library/Python/2.7/bin. I needed to patch esptool.py following <https://github.com/espressif/arduino-esp32/issues/4408> and changing the +x perms too

<https://randomnerdtutorials.com/solved-failed-to-connect-to-esp32-timed-out-waiting-for-packet-header/>

<https://github.com/me-no-dev/EspExceptionDecoder> to decode stack traces



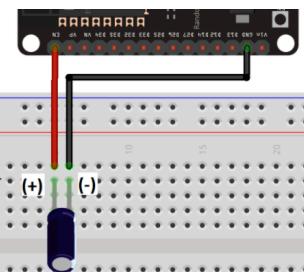
These pages came with the HiLetGo.com ESP32 Development Board

ESP32 provides the easy-of-use of the Arduino environment (forums, experts, IDE), a 32-bit 2 core, and Bluetooth and Wifi networking.

ESP32 development boards provide a breadboard-friendly package with a USB programming interface and power supply.

The downside of the development boards happens when people publish tutorials. Board change pinouts. For example, some boards provide access to ESP32's dual SPI bus pins, others do not.

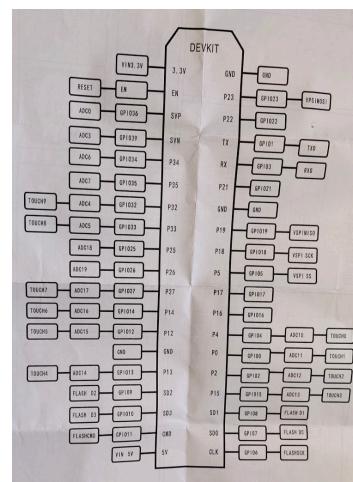
Hold the boot button down to upload a sketch



Note: electrolytic capacitors have polarity. The white/grey stripe indicates the negative lead.

If it works, then you can solder the 10 μ F electrolytic capacitor to the board. Since the EN and GND pins are far apart from each other, you can simply connect the capacitor between the EN and the GND of the ESP32 chip as shown in the schematic diagram below:

<https://randomnerdtutorials.com/solved-failed-to-connect-to-esp32-timed-out-waiting-for-packet-header/>



Wiring

Version: January 19, 2022

fcohen@starlingwatch.com, (408) 364-5508
Licensed under Creative Commons license

ESP32 Development Board				
	3.3v	19	GND	
RESET	EN	18	GPIO 23	VSPI MOSI
Left button	GPIO 36	17	GPIO 22	SCL I2C
Right button	GPIO 39	16	GPIO 01	TX0
Center/Power	GPIO 34	15	GPIO 03	RX0
GPS TX	GPIO 35	14	GPIO 21	SDA I2C
Display CS	GPIO 32	13	GND	
Backlight LED	GPIO 33	12	GPIO 19	VSPI MISO
Audio LRCL	GPIO 25	11	GPIO 18	VSPI SCK
Audio BCLK	GPIO 26	10	GPIO 5	VSPI SS
Audio DIN	GPIO 27	9	GPIO 17	Display RST
HSPI SCK	GPIO 14	8	GPIO 16	Display DC
HSPI MISO	GPIO 12	7	GPIO 4	SD memory CS
	GND	6	GPIO 0	Boot
HSPI MOSI	GPIO 13	5	GPIO 2	LED
Flash D2	GPIO 9	4	GPIO 15	HSPI SS
Flash D3	GPIO 10	3	GPIO 8	Flash D1
Flash CMD	GPIO 11	2	GPIO 7	Flash D0
	5V	1	GPIO 6	Flash SCK

Pins on ESP32 development board as seen from the top, 5V and GPIO 6 are closest to the USB connector

TX	GPS	GPS TX		
VSPI	Display	Display CS	DC	RST
VSPI	SD memory	SD memory CS		
I2S	Audio	LRCL	DIN	BCLK
I2C	Compass	SDA	SCL	
I2C	Haptic motor			
I2C	Proximity sensor			
HSPI unused				

Breakout Boards

GPS <https://www.adafruit.com/product/3133>

Compass <https://www.adafruit.com/product/2472>

Display with SD Card <https://www.adafruit.com/product/3787>

Audio <https://www.adafruit.com/product/3006>

Haptic motor <https://learn.adafruit.com/adafruit-drv2605-haptic-controller-breakout>

Proximity sensor <https://www.adafruit.com/product/466>, APDS9960

This is the wiring guide. Use it as a top-of-board legend for all the components.

The most difficult part of this project was to keep track of the multiple uses of each pin. For example, GPIO 35 is input only. GPIO 2 looks normal but has an LED off of it.

In the end all of the available GPIOs are used. GPIO 14, 12, 13, 15, 4 are used by the internal WiFi system. 9, 10, 11, 6, 7, 8 are used by the ESP32's Flash memory system (different than the SD card in this project). and 1, 3 are used by the USB adaptor.

I found conflicting tutorials, how-to guides, and references to data sheets (that turned out to be for different ESP32 development boards). One tutorial had a typo and kept referring to GPIO 20, which does not exist on the HiLetGo development board.

I also found I could not depend on translating tutorials and wiring guides for Arduino Uno and Arduino Mega to the ESP32 development boards.

Seuss uses the SD Card reader built into the Adafruit display breakout board - Display SDCS GPIO 4. I tried 2 SD Card reader boards and 2 SD cards. I could not get them to work. I have also read GPIO 4 is not available when using WiFi.

Many tutorials will reference pin numbers. While that works on Arduino Uno, the GPIO numbers are referenced here.

MEMS accelerometer, magnetometer and gyroscope

https://lcsc.com/product-detail/Attitude-Sensors_STMicroelectronics-LSM6DS3TR-C_C967633.html



Tutorial for MPU-92-65

<https://www.youtube.com/watch?v=V4NdsBjUAO0>

There are many accelerometers, compasses, and gyroscopes out there. And this board is relatively expensive. I decided on this one because it is all-in-one and requires only 2 I2C pins.

Previous versions used BNO055 and BNO085, now out of stock and maybe out of existence. <https://www.adafruit.com/product/2472>. Wiring guide at <https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/overview>, https://github.com/adafruit/Adafruit_BNO055

<https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/downloads>

Display

Version: December 23, 2021

fcohen@starlingwatch.com, (408) 364-5508
Licensed under Creative Commons license



GND
+3.3
SCL - SCK (GPIO 18)
SDA - MOSI (GPIO 23)
RES - GPIO 17
DC - GPIO 16
CS - GPIO 32
BLK - Disconnected

<https://www.buydisplay.com/download/ic/GC9A01A.pdf>
<https://www.buydisplay.com/1-28-inch-tft-ips-lcd-display-module-240x240-spi-for-arduino-raspberry-pi>
https://github.com/PaintYourDragon/Adafruit_GC9A01A

Shrink video files to no compression,

```
#!/bin/bash
cd /Users/frankcohen/Desktop/ReflectionsExperiments/ReadyForTrial
for i in *.mp4; do /Applications/ffmpeg -i "$i" -vf
"fps=15,scale=240:240:flags=lanczos,crop=240:in_h:(in_w-240)/2:0" -q:v 9 "/Users/
frankcohen/Frank/Votsh/Models/Reflections/Reflections\ Project\ Updates/Sox\
Development\ Board/Demo\ Files/10fps/${i%.}*}.mjpeg"; done

for i in *.mov; do /Applications/ffmpeg -i "$i" -vf
"fps=10,scale=240:240:flags=lanczos,crop=240:in_h:(in_w-240)/2:0" -q:v 9 "/Users/
frankcohen/Frank/Votsh/Models/Reflections/Reflections\ Project\ Updates/Sox\
Development\ Board/Demo\ Files/10fps/${i%.}*}.mjpeg"; done
```

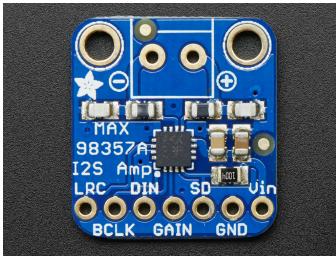
I chose this display unit because it is fast - I can get 15 frames per second - and inexpensive. It uses SPI for both the display.

GC9A01A

Audio I2S Sound Board

Version: December 23, 2021

fcohen@starlingwatch.com, (408) 364-5508
Licensed under Creative Commons license



<https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp>

libfdk_aac for converting wav to m4a (aac) is not installed by default in ffmpeg. I used these commands to install fumes with libfdk_aac on Mac OS X:

```
brew tap homebrew-ffmpeg/ffmpeg
brew install homebrew-ffmpeg/ffmpeg --with-fdk-aac --HEAD
```

Then create an AAC audio file from an mp4 video
ffmpeg -i input.wav -c:a libfdk_aac -b:a 8k output.m4a

Using this library to play I2S audio on ESP32:
<https://github.com/schreibfaul1/ESP32-audioI2S>

Audacity on Mac. Open an MP3, File -> Export, Export As MP3..., Bit rate mode: Constant, 40 KPS, save with .mp3 file extension

Use I2S audio library from
<https://github.com/schreibfaul1/ESP32-audioI2S>

Previous version used Earle F. Philhower, III,
earlephilhower@yahoo.com, for the WAV audio player library at
<https://github.com/earlephilhower/ESP8266Audio>

Use ffmpeg to create AAC audio files from mp4 video file input:
<https://trac.ffmpeg.org/wiki/Encode/AAC>

Tutorial on I2S at <https://diyi0t.com/i2s-sound-tutorial-for-esp32/>

MAX98357

The processor takes input files with:
sample rate of 44.1 kHz,
stereo format
16 bits
WAVE format

However, to play 240x240 MJPEG uncompressed files requires:
sample rate of 8000 kHz
16 bits
stereo format
WAV format
See experiments/
Stream_all_videos_audio

I use Audacity on a Mac Book Pro

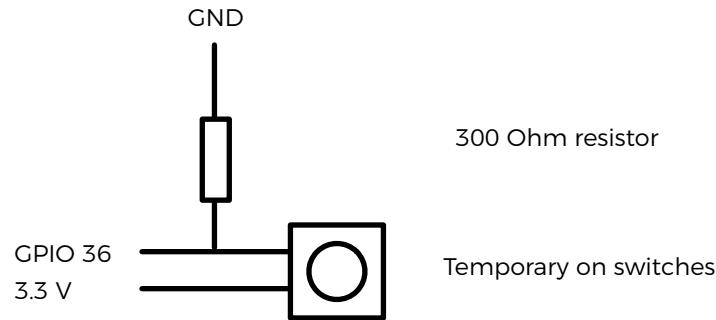
Note: Connect the Gain pin on the MAX board to GND. Do not leave it open/floating.

Buttons

Version: January 19, 2022

fcohen@starlingwatch.com, (408) 364-5508
Licensed under Creative Commons license

```
#define leftButton 34
#define centerButton 39
#define rightButton 36
```



Seuss Display includes a 3 button interface for user control. Includes a left, middle, and right button.

See the buttons.ino sketch for an example of sensing three buttons

GPIO 39 (center button) is also a control for turning the board on and off. Push and hold for 3 seconds to turn on or off.

Haptic Feedback

Version: December 23, 2021

fcohen@starlingwatch.com, (408) 364-5508
Licensed under Creative Commons license



Seuss includes a haptic feedback system using the Adafruit DRV2605 breakout board.
<https://learn.adafruit.com/adafruit-drv2605-haptic-controller-breakout>

Software is at:
https://github.com/adafruit/Adafruit_DRV2605_Library

Memory SPI SD Flash

Version: December 24, 2021

fcohen@starlingwatch.com, (408) 364-5508
Licensed under Creative Commons license



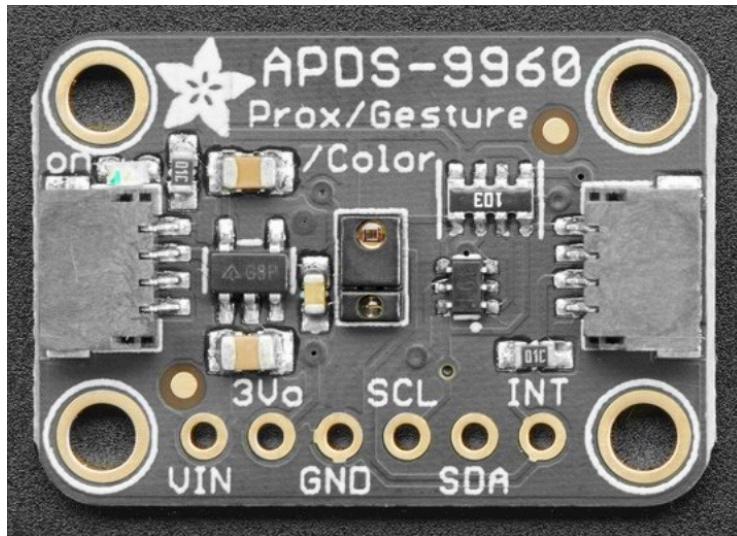
Adafruit SPI Flash SD Card - XTSD
512 MB PID: 4899

<https://www.adafruit.com/product/4899>

Gesture Sensor

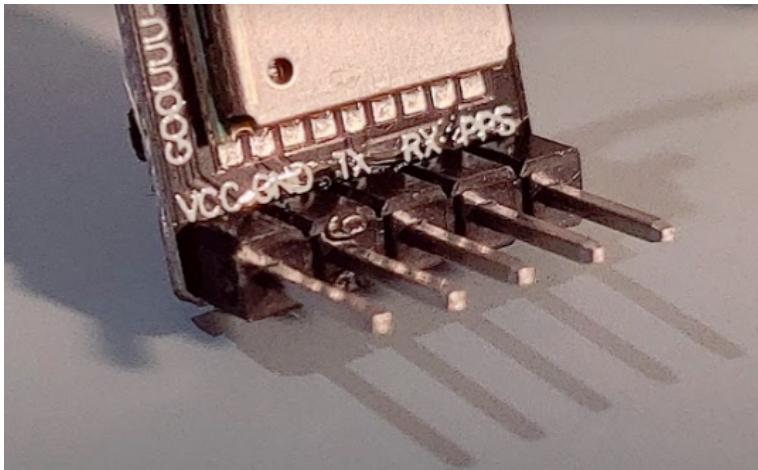
Version: December 24, 2021

fcohen@starlingwatch.com, (408) 364-5508
Licensed under Creative Commons license



<https://www.adafruit.com/product/3595>

download Adafruit_APDS9960 from the
Arduino library manager.

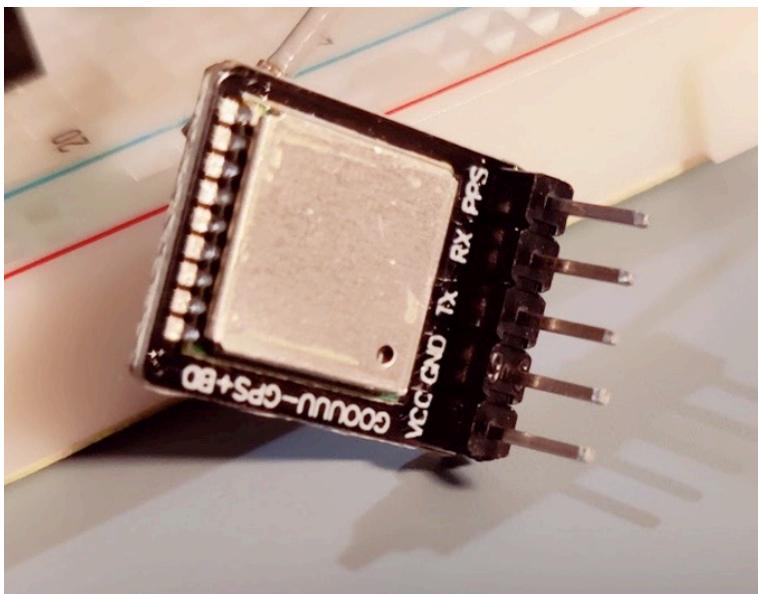


Works with 3.3 V, however, onboard LED only lights with 5 V

DFRobot, TELO132, GPS + BDS Beidou Dual Module

[https://wiki.dfrobot.com/
GPS+_BDS_Beidou_Dual_Module_SKU_TELO132](https://wiki.dfrobot.com/GPS+_BDS_Beidou_Dual_Module_SKU_TELO132)

Use TinyGPS



Use GPIO 36 (do not use 3 or 4)

[https://wiki.dfrobot.com/
GPS+_BDS_Beidou_Dual_Module_SKU_TELO132](https://wiki.dfrobot.com/GPS+_BDS_Beidou_Dual_Module_SKU_TELO132)