CST 471

Memo

To: Kevin Pintong

From: Zak Rowland

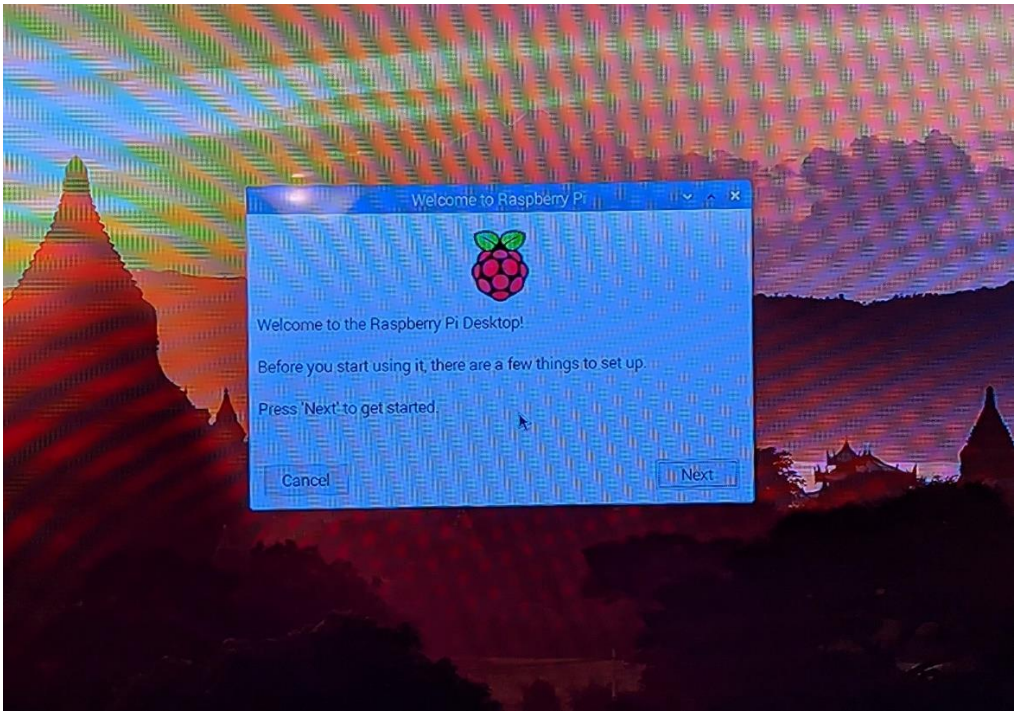
Date: December 8, 2020

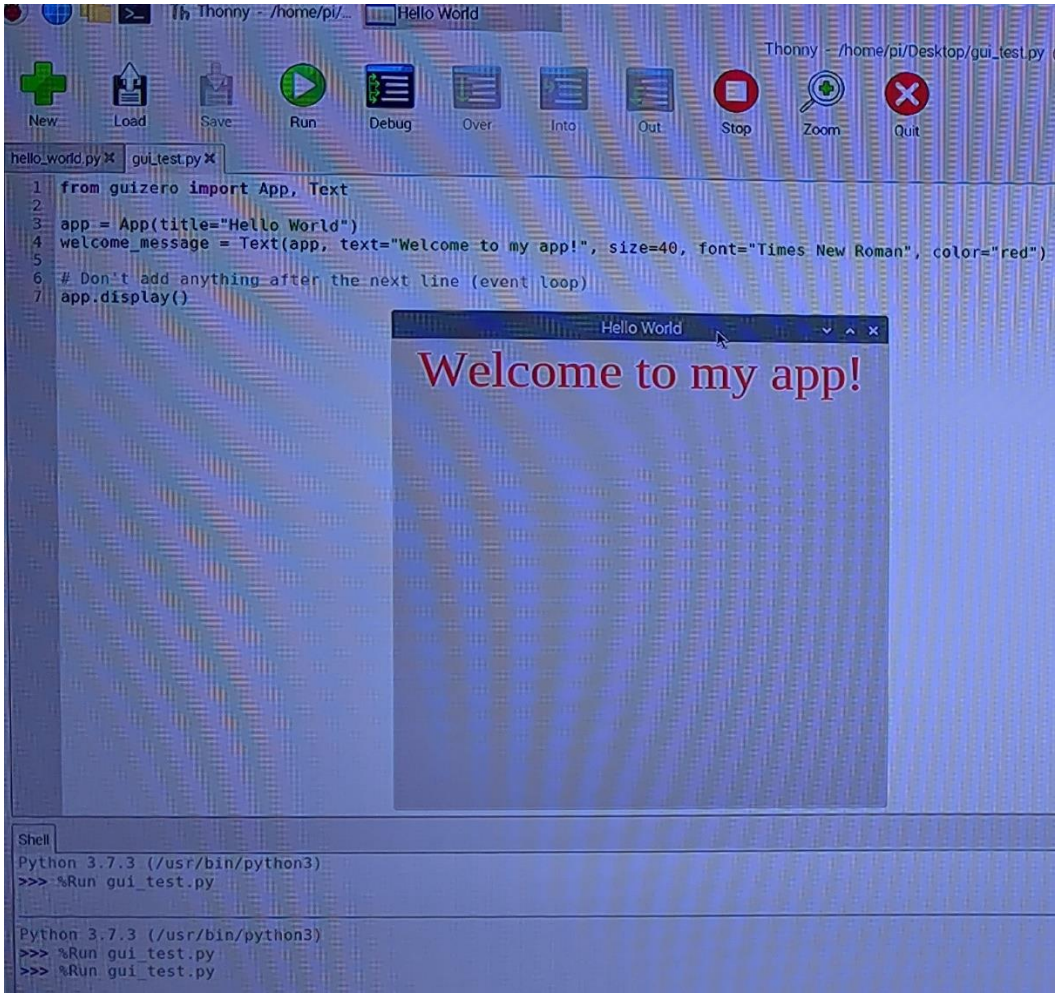
Re: Work deliverables result

Final score: 8 complete out of 12 = 66.67%

- COMPLETE. Choose an operating system for the Raspberry Pi and become familiar with it

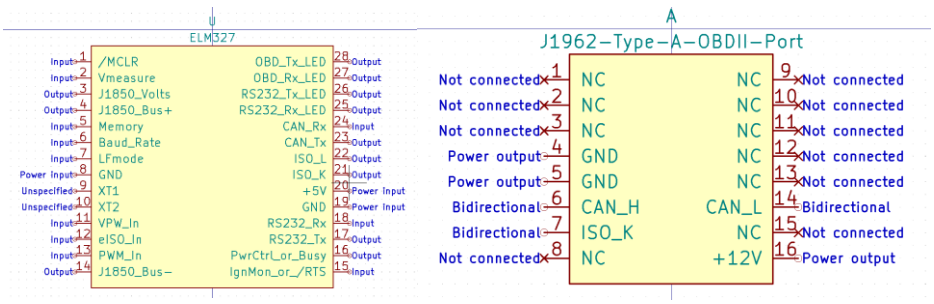
  On October 16th, I chose to install Raspberry Pi OS as the operating system. Pi OS is essentially a fork of Rasbian with the Pi Foundation's configuration on top. I chose this OS because it is recommended for new Pi users, however changing operating systems would be trivial if it is necessary. I tried out all the included applications and checked out all the configuration settings too. I also completed a very simple Python GUI tutorial to get a feel for developing a user interface. See images below.

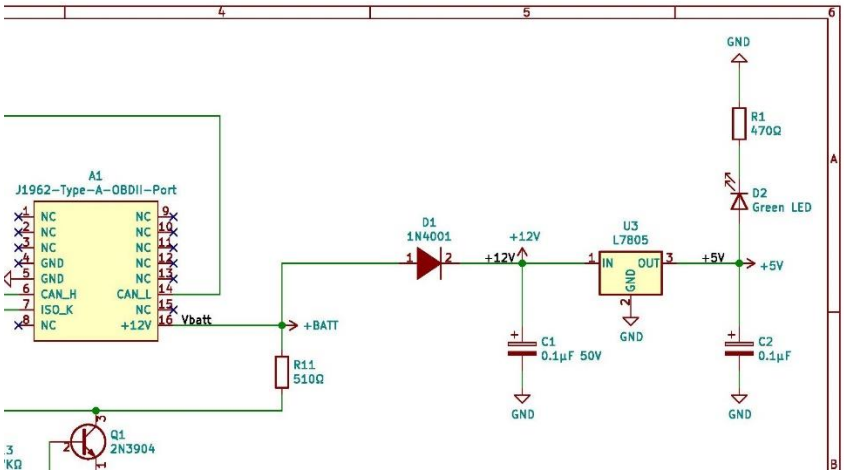




- COMPLETE. Design power circuit and develop schematic for OBD-II port power supply

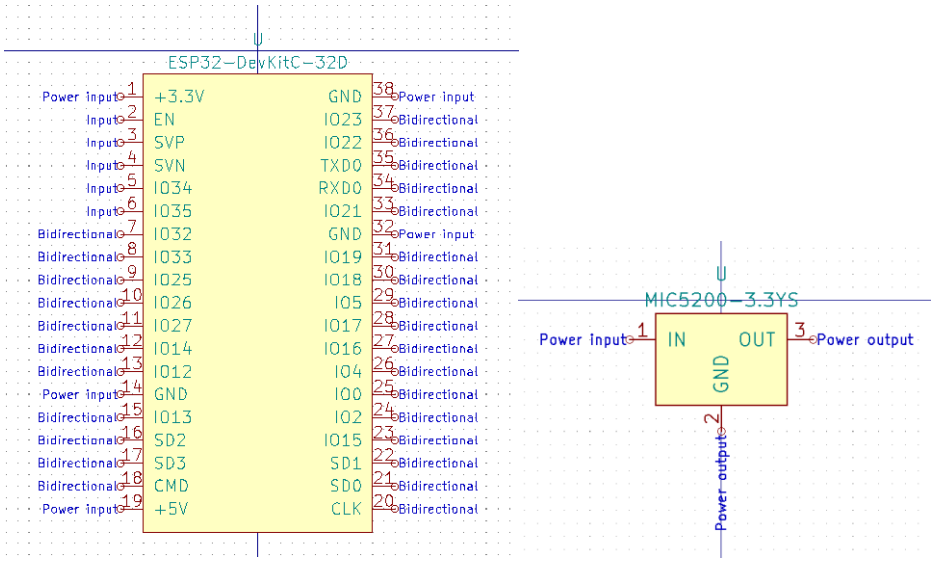
  On October 26th, progress on the OBD-II port module's schematic began. The power supply is a smaller portion of the overall schematic and was finished early on. I also started a library of symbols for any necessary symbols not included with KiCAD, this includes the ELM327 chip and the OBD-II port. See images below.

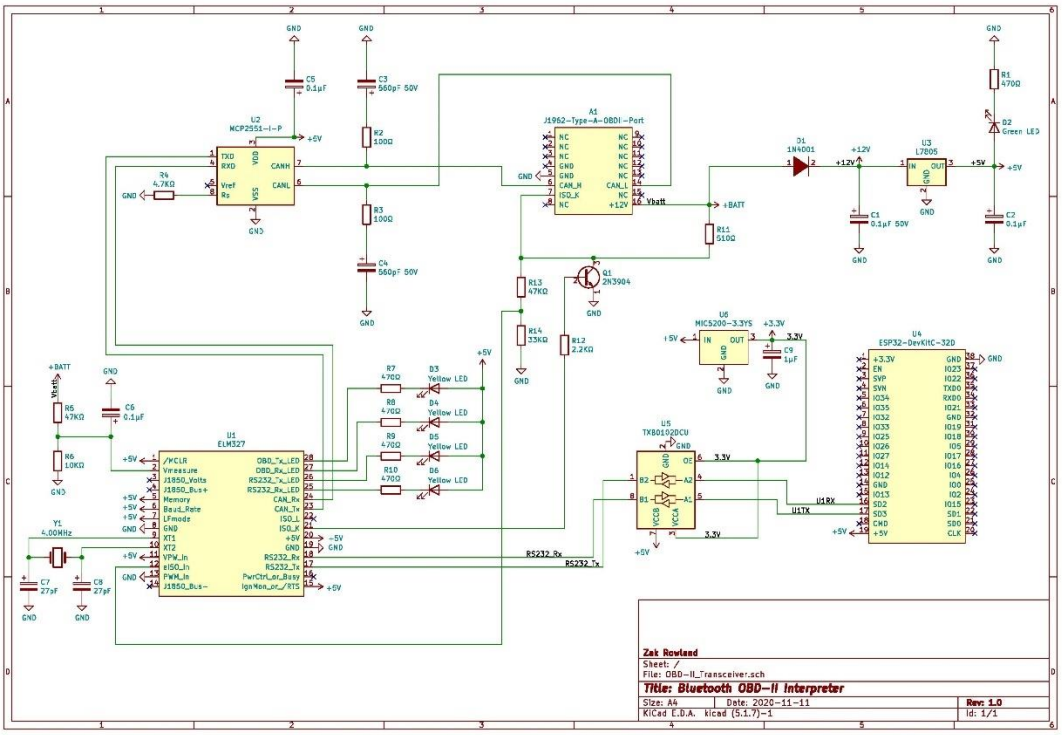




- COMPLETE. Develop schematic for OBD-II port transceiver (includes power supply mentioned above)

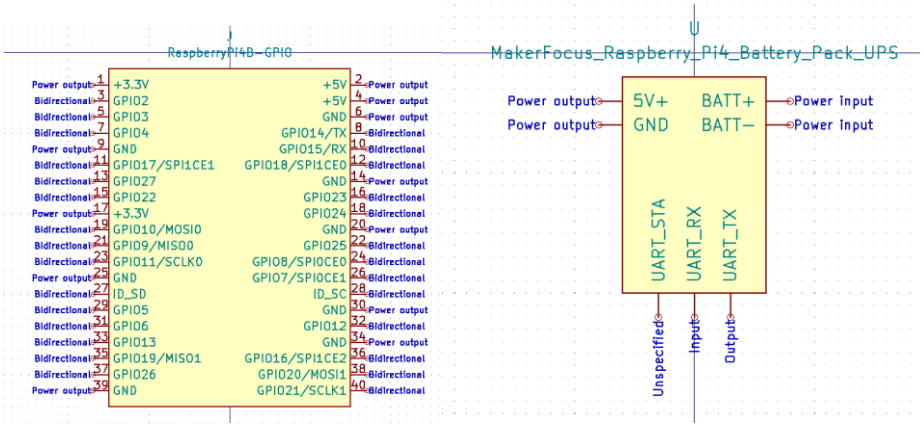
  After the power supply portion of the schematic was complete, the rest of it could be finished. One of the most difficult decisions was choosing a Bluetooth module to use. In the end it was worth researching all the different types, and the ESP32 is a perfect fit for the project. This schematic was completed on November 11th. See final OBD-II port module schematic and more symbols below.

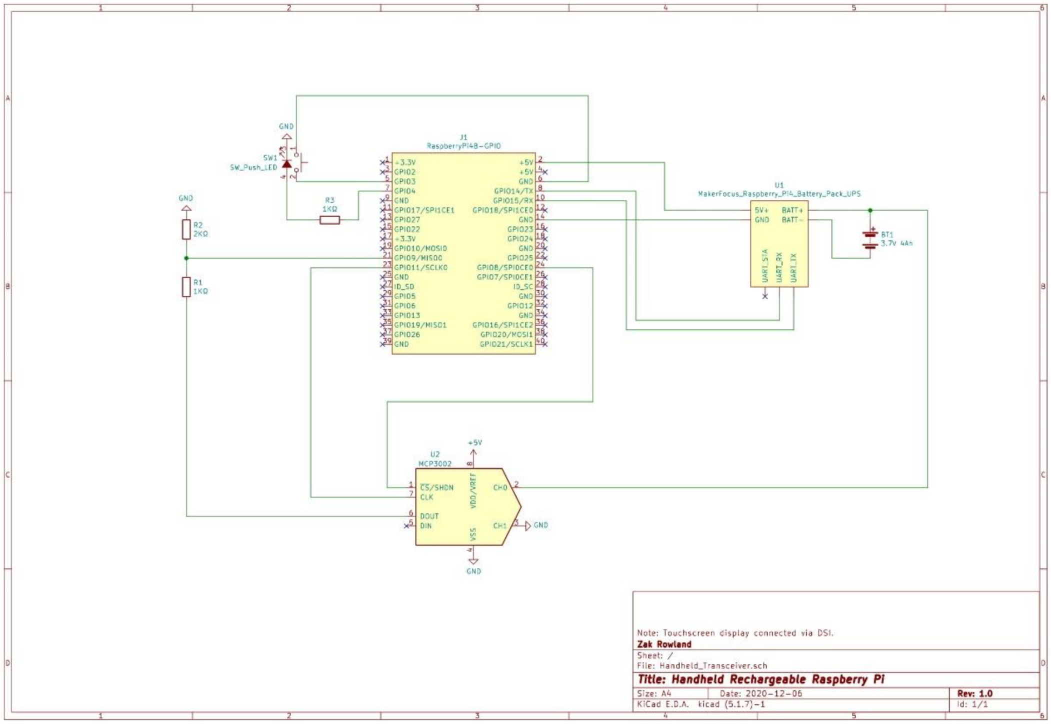




- COMPLETE. Develop schematic for handheld unit

  Progress on the handheld unit schematic began on November 11th. The most difficult part was calculating power and battery requirements and choosing a setup. I initially wanted to go with a USB-C LiPo charger and protector but I would have needed a boost to 5V or step down to 3.3V. I started looking into battery hats and found a cool UPS/battery board for the Pi that should work well. This schematic was completed December 6th. See symbols made and final schematic below.





- COMPLETE. Order necessary parts.

  I ordered all parts for OBD-II module first from Digi-Key, then once handheld unit was finished I ordered those parts from Amazon, Adafruit, and Digikey. See table of parts below.

Zak Rowland's Senior Project Part List - Last Updated 12/6/2020

Handheld Module	Part Name:	Links to Buy:	Cost Each:	Quantity:	Spares:	Shipping Cost:	Total Cost:	Purchased:	Datasheets:
	Raspberry Pi 4 (Model B)	https://www.adafruit.com/product/4292	$35.00	1	0	$0.00	$0.00	Already had.	https://www.raspberrypi.org/documentation/
	OSOYOO 5" DSI Display	https://www.amazon.com/gp/product/B07KKB5YS9	$43.88	1	0	$0.00	$43.88	Yes	https://osoyoo.com/2019/09/20/instruction-for-
	MakerFocus Raspberry Pi 4 Battery Pack UPS	https://www.amazon.com/dp/B01LAEX7J0/	$23.99	1	0	$0.00	$23.99	Yes	https://drive.google.com/file/d/12_UCy6u5iaznpg
	Momentary Switch	https://www.adafruit.com/product/3104	$1.95	1	0	$4.85	$6.80	Yes	https://cdn-shop.adafruit.com/product-
	MCP3002 ADC	https://www.digikey.com/short/zvhjjh	$1.79	1	0	$4.99	$6.78	Yes	https://ww1.microchip.com/downloads/en/Device
	1KΩ Resistor	https://www.digikey.com/short/zvhjft	$0.10	2	0	$0.00	$0.00	Already had.	https://www.seielect.com/catalog/sei-rnf_rnmf.pdf
	2KΩ Resistor	https://www.digikey.com/short/zvwbrp	$0.10	1	0	$0.00	$0.00	Already had.	https://www.seielect.com/catalog/sei-rnf_rnmf.pdf
	Samsung 32GB MicroSD	Fred Meyer	$7.99	1	0	$0.00	$7.99	Yes	N/A

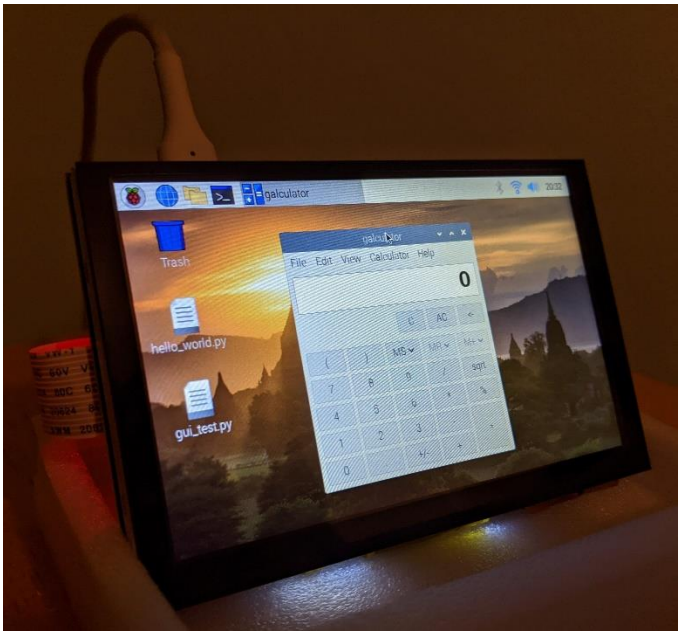
OBD-II Module	Part Name:	Links to Buy:	Cost Each:	Quantity:	Spares:	Shipping Cost:	Total Cost:	Purchased:	Datasheets:
	ELM327 (OBD-II to RS232)	https://www.elmelectronics.com/ic/elm327/	$16.00	1	0	$6.10	$22.10	Yes	https://www.elmelectronics.com/wp-
	ESP32-DevKitC-32D	https://www.digikey.com/short/zvch2v	$10.00	1	0	$10.99	$20.99	Yes	https://www.espressif.com/sites/default/files/docu
	100Ω 1/4W Resistor	https://www.digikey.com/short/zvcbd9	$0.10	2	1	$0.00	$0.30	Yes	https://www.seielect.com/catalog/sei-
	470Ω 1/4W Resistor	https://www.digikey.com/short/zvcb0v	$0.07	5	5	$0.00	$0.65	Yes	https://www.seielect.com/catalog/sei-
	510Ω 1/2W Resistor	https://www.digikey.com/short/zvcb18	$0.10	1	1	$0.00	$0.20	Yes	https://www.seielect.com/catalog/sei-cf_cfm.pdf
	2.2KΩ 1/4W Resistor	https://www.digikey.com/short/zvc570	$0.10	1	1	$0.00	$0.20	Yes	https://www.seielect.com/catalog/sei-
	4.7KΩ 1/4W Resistor	https://www.digikey.com/short/zvc537	$0.10	1	1	$0.00	$0.20	Yes	https://www.seielect.com/catalog/sei-
	10KΩ 1/4W Resistor	https://www.digikey.com/short/zvc5qo	$0.10	1	1	$0.00	$0.20	Yes	https://www.seielect.com/catalog/sei-
	33KΩ 1/4W Resistor	https://www.digikey.com/short/zvc582	$0.10	1	1	$0.00	$0.20	Yes	https://www.seielect.com/catalog/sei-
	47KΩ 1/4W Resistor	https://www.digikey.com/short/zvc5j0	$0.10	2	1	$0.00	$0.30	Yes	https://www.seielect.com/catalog/sei-
	27pF 50V Capacitor	https://www.digikey.com/short/zvcf0n	$0.22	2	1	$0.00	$0.66	Yes	https://www.vishay.com/docs/45171/kseries.pdf
	560pF 50V Capacitor	https://www.digikey.com/short/zvc27m	$0.24	2	1	$0.00	$0.72	Yes	https://product.tdk.com/info/en/catalog/datashee
	0.1uF 50V Capacitor	https://www.digikey.com/short/zvc28j	$0.10	4	1	$0.00	$0.50	Yes	https://katalog.we-
	1uF 50V Capacitor	https://www.digikey.com/short/zvc2jf	$0.10	1	1	$0.00	$0.20	Yes	https://katalog.we-
	1N4001 Diode	https://www.digikey.com/short/zvc245	$0.10	1	1	$0.00	$0.20	Yes	https://www.comchiptech.com/admin/files/produc
	Green LED	https://www.digikey.com/short/zvc29p	$0.33	1	1	$0.00	$0.00	Already had.	http://www.kingbrightusa.com/images/catalog/SP
	Yellow LED	https://www.digikey.com/short/zvc2f7	$0.21	4	1	$0.00	$0.00	Already had.	https://katalog.we-
	2N3904 NPN	https://www.digikey.com/short/zvc2dt	$0.20	1	1	$0.00	$0.40	Yes	https://www.onsemi.com/pub/Collateral/PZT3904-
	MCP2551 Transceiver	https://www.digikey.com/short/zvc204	$1.09	1	0	$0.00	$1.09	Yes	http://www.microchip.com/mymicrochip/filehandle
	L7805 Regulator	https://www.digikey.com/short/zvc2v9	$0.50	1	1	$0.00	$1.00	Yes	http://www.st.com/content/ccc/resource/technic
	TXB0102DCUR	https://www.digikey.com/short/zvcw72	$0.71	1	0	$0.00	$0.71	Yes	http://www.ti.com/general/docs/suppproductinfo.t
	MIC5200-3.3	https://www.digikey.com/short/zvcwtv	$1.03	1	0	$0.00	$1.03	Yes	http://www.microchip.com/mymicrochip/filehandle
	4.00MHz crystal	https://www.digikey.com/short/zvcw3m	$0.66	1	0	$0.00	$0.66	Yes	http://www.ecsxtal.com/store/pdf/hc-49usx.pdf
	J1962-TypeA connector	https://www.ebay.com/itm/DIY-Universal-16-Pin-OBD2-	$2.49	1	0	$0.00	$2.49	Yes	N/A

- INCOMPLETE. Test the ELM327 chip (there is a command to issue that will return the version number if everything is working)

  Testing the ELM327 requires most of the circuitry to be assembled already, and I do not have a multimeter yet to test all my components before connecting them. It was an oversight from the beginning of the term; I thought I could get more done than I really could. Now that I have actually made progress on the project, I will have be able to set more realistic goals going forward.

- COMPLETE. Connect touchscreen to Raspberry Pi and register a touch

  The DSI display I chose was truly plug and play; I was able to boot up the Pi with the screen connected and navigate the OS by touch right away. A demo of the touch screen was provided during the design review, and see the image below of it connected.

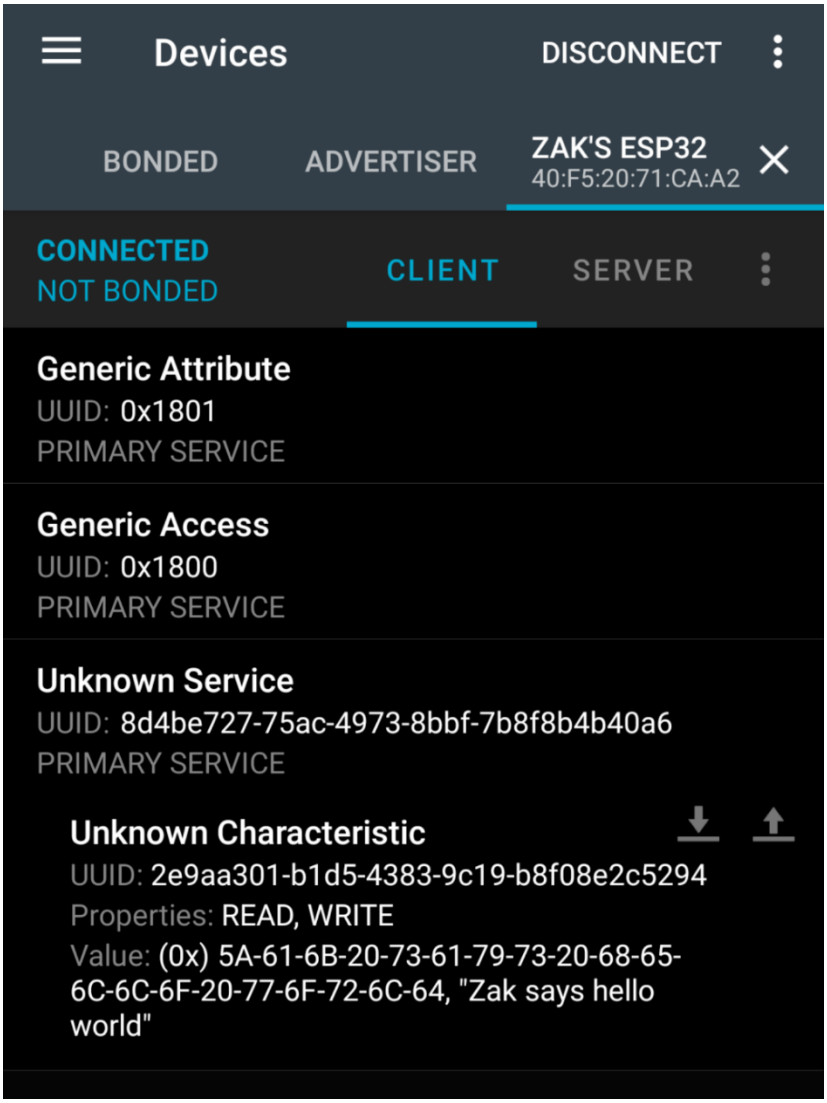


- COMPLETE. Configure the Bluetooth module (changing name, PIN, mode, etc.)

  For the final design I decided to use Visual Studio Code with the official Espressif ESP32 extension. This provides more control over the hardware and includes functionality like a debugger and terminal. For the purposes of testing the Bluetooth functionality I used the Arduino IDE since the included Bluetooth examples are far less complex. This was completed December 5th. I used my phone to verify the configuration worked, see images below.

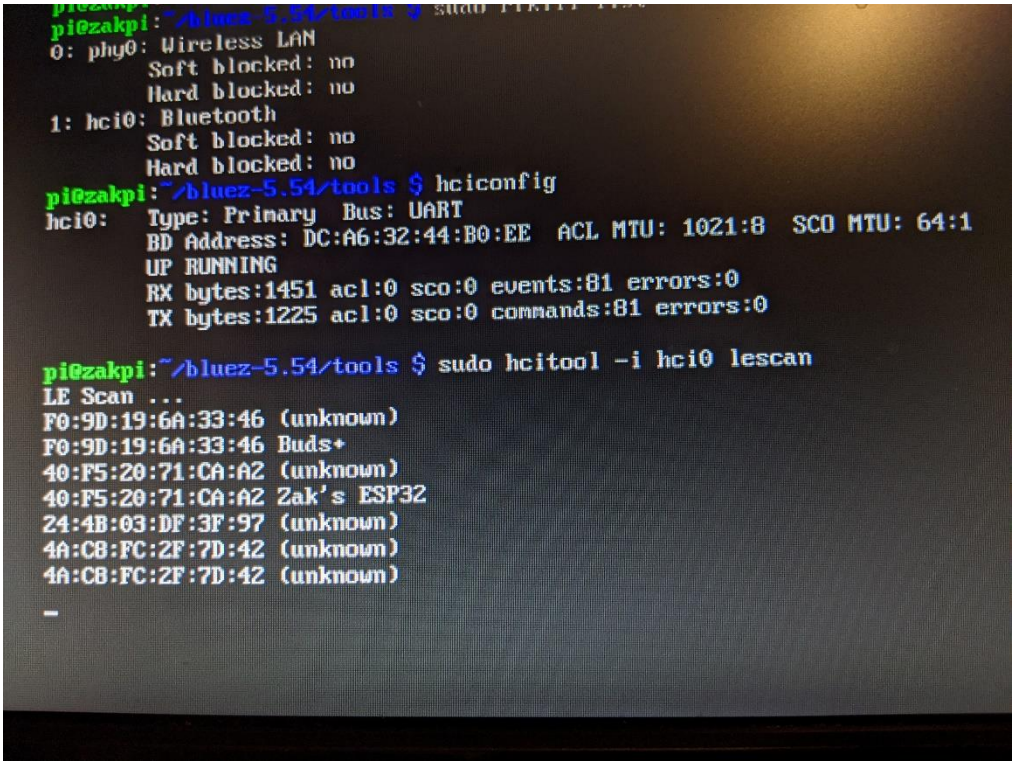
BLE_server §

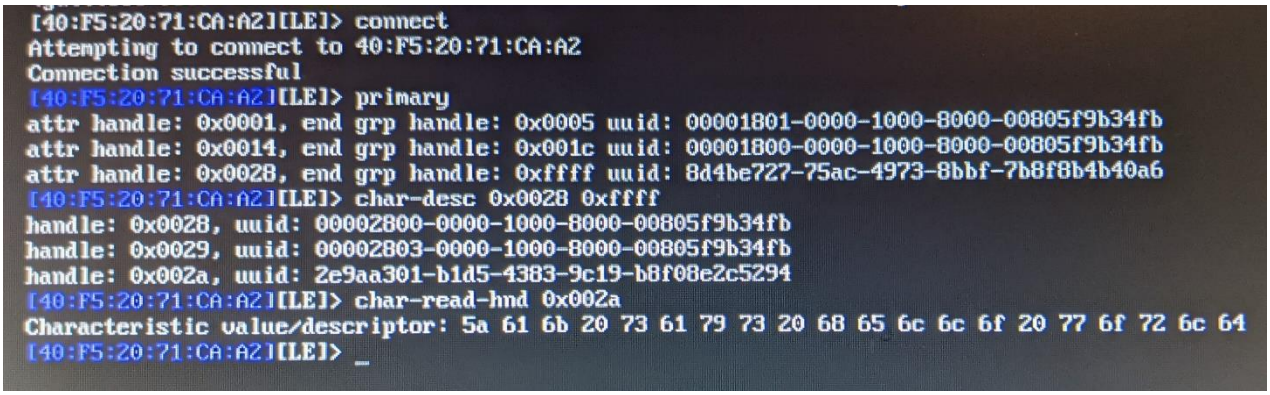
```
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEServer.h>

// See the following for generating UUIDs:
// https://www.uuidgenerator.net/

#define SERVICE_UUID        "8d4be727-75ac-4973-8bbf-7b8f8b4b40a6" // Generated 12/5/2020
#define CHARACTERISTIC_UUID "2e9aa301-b1d5-4383-9c19-b8f08e2c5294" // Generated 12/5/2020

void setup() {
  Serial.begin(115200);
  Serial.println("Starting BLE work!");

  BLEDevice::init("Zak's ESP32");
  BLEServer *pServer = BLEDevice::createServer();
  BLEService *pService = pServer->createService(SERVICE_UUID);
  BLECharacteristic *pCharacteristic = pService->createCharacteristic(
                                         CHARACTERISTIC_UUID,
                                         BLECharacteristic::PROPERTY_READ |
                                         BLECharacteristic::PROPERTY_WRITE
                                       );

  pCharacteristic->setValue("Zak says hello world");
  pService->start();
  // BLEAdvertising *pAdvertising = pServer->getAdvertising();  // this still is working for backward compatibility
  BLEAdvertising *pAdvertising = BLEDevice::getAdvertising();
  pAdvertising->addServiceUUID(SERVICE_UUID);
  pAdvertising->setScanResponse(true);
```



- COMPLETE. Establish a Bluetooth connection between Raspberry Pi and Bluetooth module

  The Pi doesn't include a Bluetooth stack out of the box, so I download BlueZ since it seemed to be popular. After some troubleshooting, I was able to see the ESP32 from the Pi and connect to it. The same characteristic value can be read (except in hex only) on the Pi that the phone read above ("Zak says hello world"). See images below.





- INCOMPLETE. Successfully read data from the OBD-II port over Bluetooth

  Testing the ELM327 requires most of the circuitry to be assembled already, and I do not have a multimeter yet to test all my components before connecting them. It was an oversight from the beginning of the term; I thought I could get more done than I really could. Now that I have actually made progress on the project, I will have be able to set more realistic goals going forward.

- INCOMPLETE. Successfully interpret data from the OBD-II port over Bluetooth

  See above.

- INCOMPLETE. Successfully write data to the OBD-II port over Bluetooth

  See above.