# FINAL PRESENTATION

Zak Rowland

CST 473 – Embedded Senior Project

# WHAT IS IT?

- Bluetooth OBD-II diagnostic tool with rechargeable touchscreen handheld
  - Diagnose check engine lights or other malfunctions
  - Monitor speed, temperatures, and pressures
  - Perfect for home mechanics or hobbyists, professionals too
  - Low cost and open source


Source

# REQUIREMENTS

- The Bluetooth connection and GUI was not finished
  - This leaves a few requirements unmet
- Successful communication with the car through the hardware was a big milestone
- 10.33 of 14 requirements met, or 73.8%

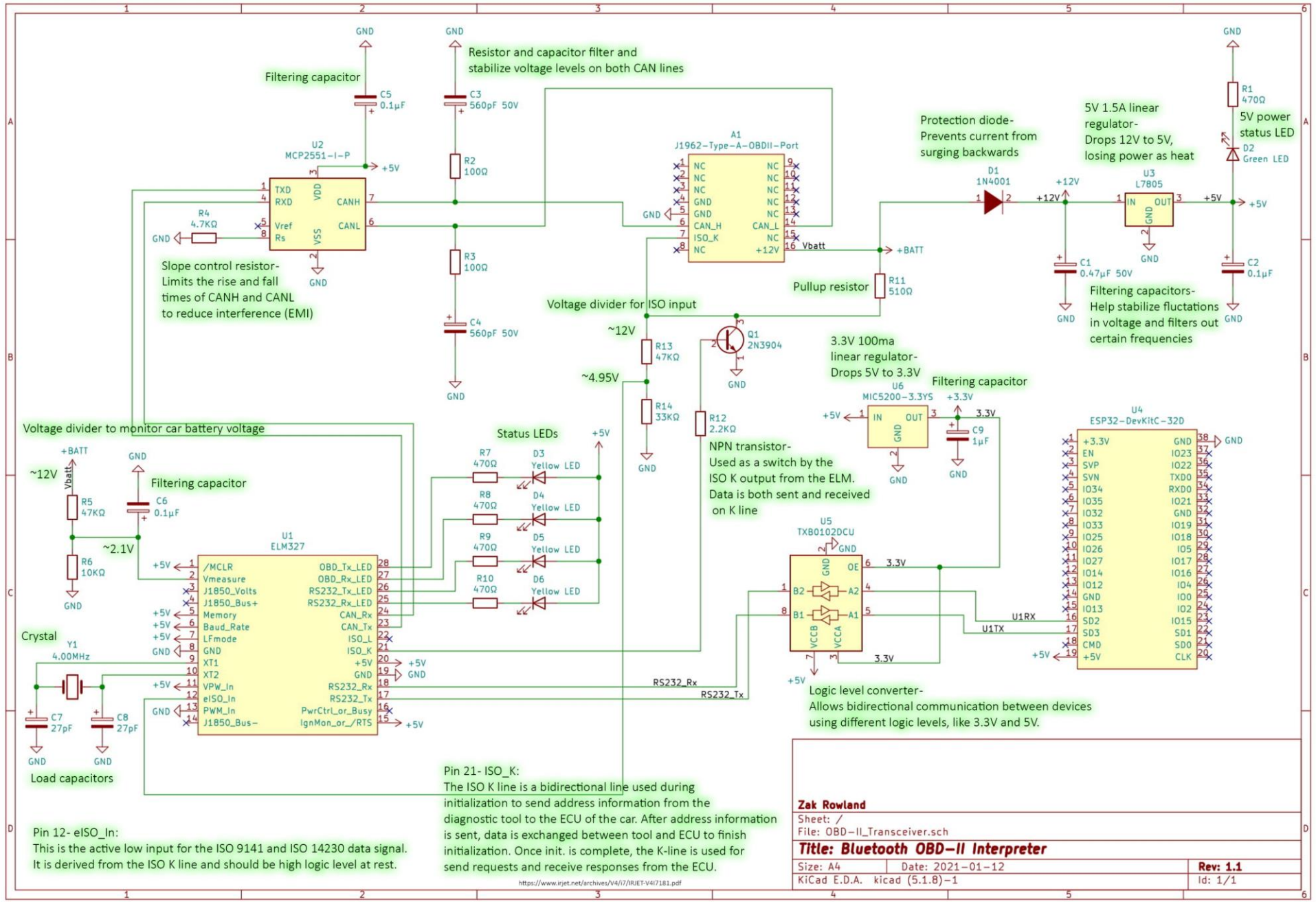
# REQUIREMENTS CONTINUED

- The most significant unmet requirements are:

1. The system shall be able to read and clear diagnostic (trouble) codes.
    a. The user interface will display the diagnostic codes in list form with buttons to scroll up and down through the list.
        i. The list will display the diagnostic codes (e.g. P0011) only.
        ii. The user must touch one of the diagnostic codes to read the description or possible cause.
    b. The user interface will provide a button to clear all diagnostic codes.
2. The system shall have the ability to read sensor data at minimum 30 times per second including speed, coolant temperature, and oil pressure.
    a. The user interface will display the data in decimal format.
        i. The option for digital gauges may be implemented.
    b. The data will be displayed *by default* in units of miles per hour for speed, Fahrenheit for temperature, and pounds per square inch for pressure.
        i. The option for metric units shall be implemented.

# COST AND TIMELINE

- Original NRE cost estimate was $53,053
  - Includes salary, breadboards, power supply, multimeter, oscilloscope, and HDMI cable
- Original production cost estimate was $171.53
  - Includes Raspberry Pi, ESP32, ICs, touchscreen, components, etc.
- Final NRE cost is $53,060.87
  - Needed surface mount to dual inline pin adapters, 15V power supply
- Final production cost is $157.28


Filtering capacitor
C5 0.1µF
GND
+5V
U2
MCP2551-I-P
TXD
RXD
Vref
Rs
VDD
VSS
CANH
CANL
R4 4.7KΩ
Slope control resistor-
Limits the rise and fall times of CANH and CANL to reduce interference (EMI)
Resistor and capacitor filter and stabilize voltage levels on both CAN lines
C3 560pF 50V
R2 100Ω
R3 100Ω
C4 560pF 50V
A1
J1962-Type-A-OBDII-Port
NC
GND
CAN_H
CAN_L
ISO_K
+12V
Vbatt
+BATT
Protection diode-
Prevents current from surging backwards
D1
1N4001
5V 1.5A linear regulator-
Drops 12V to 5V, losing power as heat
U3
L7805
IN
OUT
C1 0.47µF 50V
C2 0.1µF
Filtering capacitors-
Help stabilize fluctations in voltage and filters out certain frequencies
R1 470Ω
5V power status LED
D2
Green LED
Pullup resistor
R11 510Ω
Voltage divider for ISO input
~12V
~4.95V
R13 47KΩ
R14 33KΩ
R12 2.2KΩ
Q1
2N3904
NPN transistor-
Used as a switch by the ISO K output from the ELM. Data is both sent and received on K line
3.3V 100ma linear regulator-
Drops 5V to 3.3V
U6
MIC5200-3.3YS
C9 1µF
Filtering capacitor
3.3V
Voltage divider to monitor car battery voltage
R5 47KΩ
R6 10KΩ
~2.1V
C6 0.1µF
Filtering capacitor
U1
ELM327
/MCLR
Vmeasure
J1850_Volts
J1850_Bus+
Memory
Baud_Rate
LFmode
XT1
XT2
VPW_In
eISO_In
PWM_In
J1850_Bus-
OBD_Tx_LED
OBD_Rx_LED
RS232_Tx_LED
RS232_Rx_LED
CAN_Rx
CAN_Tx
ISO_L
ISO_K
RS232_Rx
RS232_Tx
PwrCtrl_or_Busy
IgnMon_or_/RTS
Status LEDs
R7 470Ω
R8 470Ω
R9 470Ω
R10 470Ω
D3 Yellow LED
D4 Yellow LED
D5 Yellow LED
D6 Yellow LED
Crystal
Y1
4.00MHz
C7 27pF
C8 27pF
Load capacitors
U5
TXB0102DCU
OE
B2
B1
A2
A1
VCCB
VCCA
U1RX
U1TX
RS232_Rx
RS232_Tx
Logic level converter-
Allows bidirectional communication between devices using different logic levels, like 3.3V and 5V.
U4
ESP32-DevKitC-32D
Pin 12- eISO_In:
This is the active low input for the ISO 9141 and ISO 14230 data signal. It is derived from the ISO K line and should be high logic level at rest.
Pin 21- ISO_K:
The ISO K line is a bidirectional line used during initialization to send address information from the diagnostic tool to the ECU of the car. After address information is sent, data is exchanged between tool and ECU to finish initialization. Once init. is complete, the K-line is used for send requests and receive responses from the ECU.
https://www.irjet.net/archives/V4/i7/IRJET-V4I7181.pdf
Zak Rowland
Sheet: /
File: OBD-II_Transceiver.sch
Title: Bluetooth OBD-II Interpreter
Size: A4
Date: 2021-01-12
Rev: 1.1
KiCad E.D.A. kicad (5.1.8)-1
Id: 1/1

# DEMO

# LESSONS LEARNED

- Working remotely made the project difficult to work on
  - I should have finished little pieces every day instead of long stretches of work
  - Reach out for help more
- Don’t waste time on things that aren’t important
  - Choosing an overly complicated IDE
  - Playing around with Raspberry Pi operating systems

# THE END