Self-Learning Assignment:
Creating Custom KiCad Symbols and Footprints

Joseph Serra
934-512-465
Oregon State University
ENGR 416 – Multidisciplinary Capstone Design

January 26, 2026

1 Introduction

For my self-learning assignment, I learned how to create custom schematic symbols and PCB footprints in KiCad. My capstone project requires the HUSB238A, a USB-PD sink controller from Hynetek that allows devices to negotiate specific voltage and current levels from USB-C power sources. This component was not available in existing KiCad libraries, so I needed to create my own symbol and footprint to proceed with the PCB design.

I used three main resources for learning. The official KiCad documentation [1] covered the Symbol Editor and Footprint Editor interfaces. PCBway's tutorial [2] walked through footprint creation step-by-step. The Embedded Systems Design guide [3] was particularly useful for understanding how to interpret datasheet dimensions and translate them into pad placements.

2 Application

2.1 Creating the Schematic Symbol

I started by opening KiCad's Symbol Editor and creating a new library for my capstone project components. Following the conventions from the documentation, I set the grid to 50 mils (1.27 mm) to ensure pins would align properly in schematics.

I then created a new symbol and added the rectangular body outline. The HUSB238A has 11 pins, which I added one at a time with the correct names, numbers, and electrical types based on the datasheet:

- VIN (Pin 1) – Power input from USB-C VBUS
- D+/D- (Pins 2–3) – USB data lines for BC1.2 detection
- CC1/CC2 (Pins 4–5) – USB-C configuration channel pins
- SDA/SCL (Pins 6–7) – I2C interface for configuration
- VSET/ISET (Pins 8–9) – Voltage and current setting pins
- GATE (Pin 10) – External MOSFET gate driver output
- GND (Pin 11) – Ground connection via exposed thermal pad

I placed power pins (VIN, GND) on the left side and signal pins on the right, following standard symbol conventions. I also added a pin-1 indicator dot and configured the reference designator field.

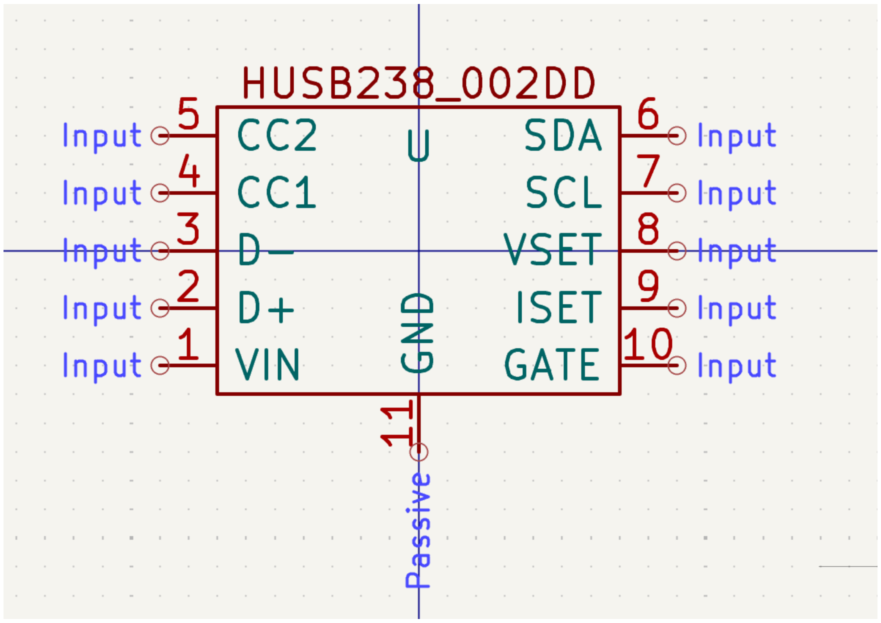
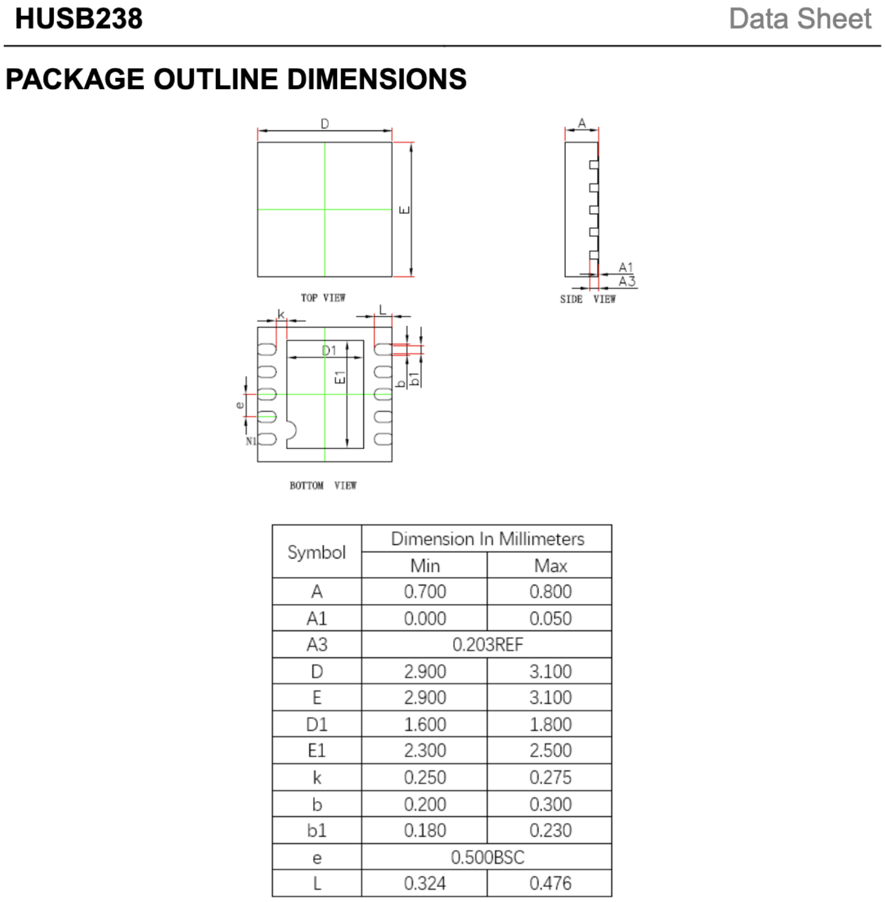


Figure 1: Custom schematic symbol for the HUSB238A in KiCad's Symbol Editor.

2.2 Creating the PCB Footprint

The footprint was more involved since it required translating physical dimensions from the datasheet into pad locations. The HUSB238A uses a DFN-10 package with an exposed thermal pad on the bottom.



HUSB238 Data Sheet

PACKAGE OUTLINE DIMENSIONS

Symbol	Dimension In Millimeters	
	Min	Max
A	0.700	0.800
A1	0.000	0.050
A3	0.203REF	
D	2.900	3.100
E	2.900	3.100
D1	1.600	1.800
E1	2.300	2.500
k	0.250	0.275
b	0.200	0.300
b1	0.180	0.230
e	0.500BSC	
L	0.324	0.476

Figure 2: Package outline dimensions from the HUSB238A datasheet showing the DFN package specifications.

From the datasheet dimension table, I extracted the values needed for the footprint:

- Pin pitch (e): 0.500 mm – the center-to-center distance between adjacent pins
- Pad width (b): 0.20–0.30 mm – I used 0.25 mm as a nominal value
- Pad length (L): 0.324–0.476 mm – I used 0.40 mm to ensure good solder contact
- Package body (D × E): approximately 3.0 mm × 3.0 mm
- Exposed thermal pad (D1 × E1): 1.6–1.8 mm × 2.3–2.5 mm

In the Footprint Editor, I switched the grid units to millimeters to match the datasheet. I created each SMD pad with the dimensions calculated above, positioning them according to the pin pitch. The most important step was ensuring the pin numbering matched exactly between the symbol and footprint – if pin 1 is VIN on the symbol, it must also be VIN on the footprint, or the PCB will be incorrect.

For the exposed thermal pad, I created a larger rectangular pad in the center and assigned it as Pin 11 to match the symbol's GND pin. I added a silkscreen outline slightly larger than the package body and included a courtyard boundary for design rule checking.

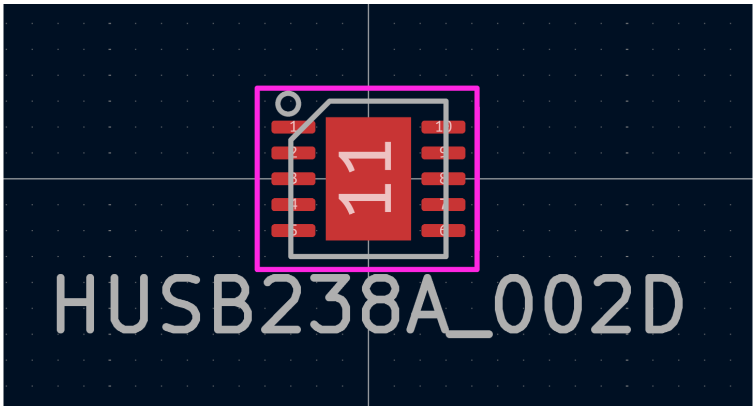


Figure 3: Completed footprint showing 10 SMD pads and the central thermal pad (Pin 11).

2.3 Integration and Testing

After completing both components, I linked the symbol to the footprint using KiCad's symbol properties dialog. I then tested the components by placing the symbol in my capstone project's USB-C Power Delivery schematic. The schematic in Figure 4 shows the HUSB238A connected to a USB-C connector, with the CC1/CC2 pins handling USB-PD communication and the GATE pin driving an external MOSFET that switches the power path. This circuit will allow our capstone device to request up to 20V from USB-PD power sources.

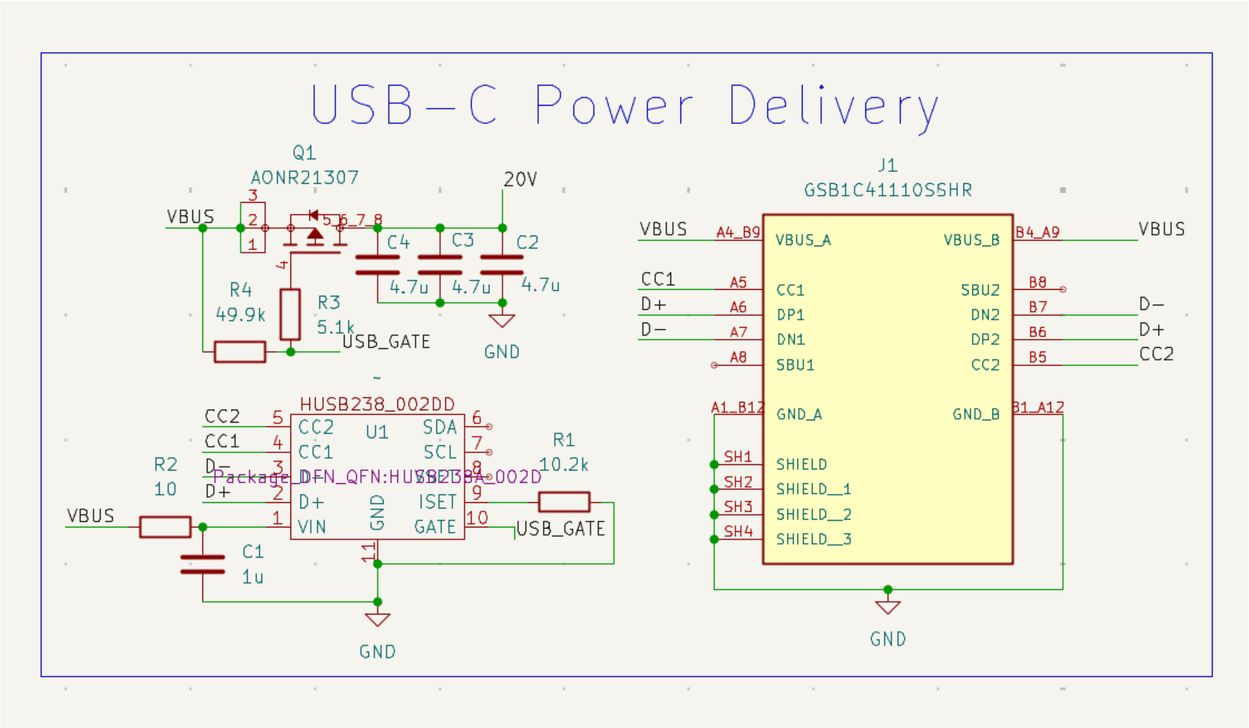


Figure 4: The HUSB238A symbol integrated into the USB-C Power Delivery schematic.

3 Conclusion

This self-learning exercise gave me practical experience with KiCad's component creation tools. The main takeaway was the importance of carefully reading datasheets – small errors in dimension interpretation can result in a footprint that doesn't match the physical component. I now feel comfortable creating custom symbols and footprints for other components our project may need.

References

[1] KiCad Documentation Team, "Getting Started in KiCad," KiCad 8.0 Documentation. [Online]. Available: `https://docs.kicad.org/8.0/en/getting_started_in_kicad/getting_started_in_kicad.html`

[2] PCBway, "How to make a footprint in KiCad?," PCB Design Tutorial. [Online]. Available: `https://www.pcbway.com/blog/PCB_Design_Tutorial/How_to_make_a_footprint_in_KiCad_.html`

[3] Embedded Systems Design, "Using the KiCad Footprint Editor," Arizona State University. [Online]. Available: `https://embedded-systems-design.github.io/kicad-footprint-editor/`