Mini Digital Multimeter Project

Fall 2025 Report

Project Team:

Peter Kocour

Ian Ranstadler

Nathan Campbell

Table of Contents

1. Executive Summary

The Mini Digital Multimeter Project presents a compact measurement device, about the size of a credit card, which is capable of capturing DC voltage measurements of up to 30 volts and DC current measurements of up to 1 amp. It’s designed with affordability and mass production in mind, maintaining a bill of materials cost of roughly $10 per unit at a scale of 100 units.

Core Technology

The Digital Multimeter utilizes the ATmega328P microcontroller, common on arduino platforms, supplemented by a four circuit op-amp, a charge pump, a display driver, and a 4 digit display. The four circuit op-amp provides a buffer for the inputs of the ATmega internal ADC for voltage measurements and provides amplification for the current measurements. The op-amp is provided with both a positive 3.3 volt supply via the battery and a negative 3.3 volt supply via the charge pump. The measured voltages and currents are then displayed through the display driver and onto the 4 digit display.

Innovations

Several key innovations were integrated in the mini digital multimeter, from a unique display and display driver combo to a single op-amp floating shunt resistor circuit measurement circuit, which pushed the limits of accuracy while maintaining a low overall bill of material cost. The display driver and 4 digit display proved to be the most challenging, requiring a deep dive into the display driver datasheet and the C++ code to determine how to get the correct segments to turn on and off on the display based on a collected measurement. Another innovation was the addition of a charge pump to provide the required negative 3.3 volts for the supply rail of the op-amp with the addition of a few capacitors. For current measurement, it was discovered that the change in voltage drop across a small shunt resistor was too small for the internal ADC to see, thus requiring some sort of amplification to make it easier to identify changes in voltages and thus changes in currents.

Challenges and Solutions

Our greatest challenge with the project stemmed from the display and display driver IC. One critical part of the project was keeping costs low. The display on the previous design was not only physically bulky but pretty pricey for our goal of a small, low cost device. Moving away from the LCD to a cheaper alternative proved limiting and challenging. Most displays were not cheap, often in the range of $2 to $5, much more than we would like to spend. As a solution, we found the LCD-S401M16KR by Lumex which was at most $1.67 at low quantities, but more importantly, the price dropped to $0.8991 when ordering a quantity of 100. While this initially solved our first problem, it created a second, driving the display that wants an AC voltage signal. One potential solution was using a PWM signal from the microcontroller pins to see if that would be enough to turn on and off the display. During testing this solution almost worked, however, it took up all of the GPIO pins, saving nothing for the rest of the devices, like LEDs, switches, etc, we needed to use. This ultimately led to the use of the BU91796FS-ME2 display driver IC. This device could be communicated with over I2C, would handle all of the AC signal, 1/4 duty, 1/3 bias, requirements of the display and only cost $1.1022, putting our total cost at just around $2 being one of the cheapest options. Pawning off the control of the display to this driver frees up many of the GPIO pins on the microcontroller while also handling all of the challenges with driving the display. That being said, this solution still introduced another problem, finding the right I2C commands to turn on and off the current segments on the display when we need or want to. Programming the correct commands was a challenge that didn't get solved at the end of the semester. While some of the segments turn on when commanded displaying the numbers 1, 4, 0, 5, and a few others correctly, not every segment turns on when needed, preventing every number from being displayed correctly. With some more time spent in the datasheet for the display driver and the I2C commands, the correct commands could eventually be determined.

Impact

The Open Source MINI Digital Multimeter offers a hands-on give-away item to future MTU students or others interested in electronics. This device will give its users a hands-on education of electronics, promoting technical development at a fraction of the cost of a commercial multimeter. By open-sourcing hardware and software, this project encourages students, makers, and educators to innovate, customize, and better understand electronics and electrical tools.

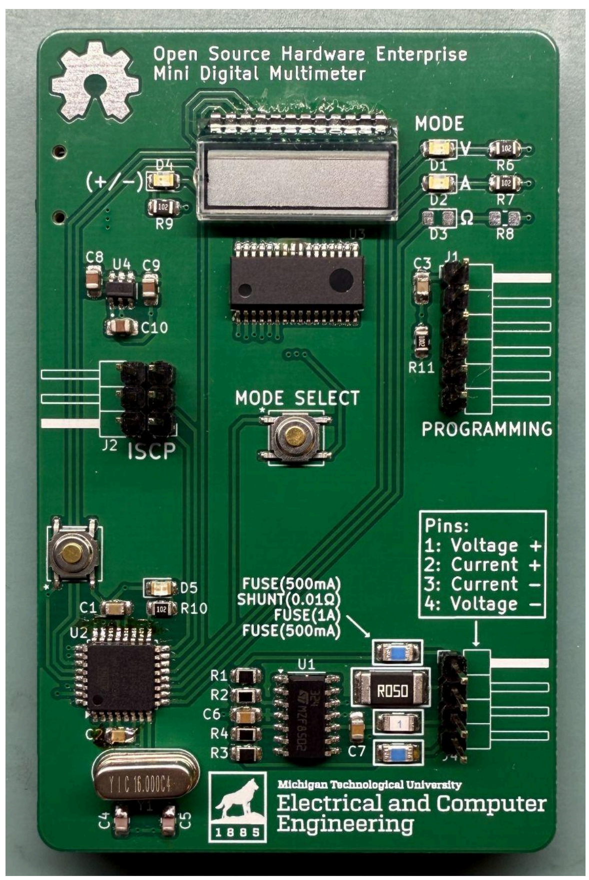

Figure 1: Mini DMM Version 1 Product Image

2. Project Description

The Mini Digital Multimeter Project presents a compact measurement device, about the size of a credit card, which is capable of capturing DC voltage measurements of up to 30 volts, DC current measurements of up to 1 amp, and resistance measurements up to 100,000 ohms. Designed with affordability and mass production in mind, the device will maintain a bill of materials cost of roughly $10 per unit at a scale of 100 units and prioritize surface mount components while minimizing through hole components where possible. Switching between measurement modes will be easy and intuitive, with a clear mode selection button and indicators to guide users to their desired measurement type. This device will also employ numerous safety features such as fuses, graphical warnings and an enclosure to protect users and ensure safe operation.

3. Methodology

3.1 Voltage Measurement

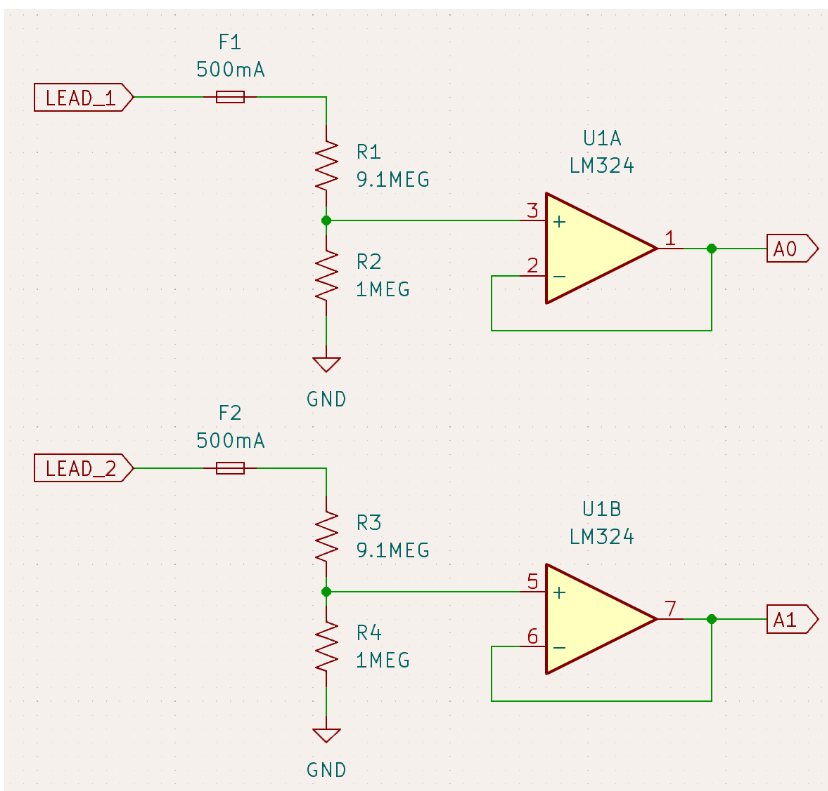

Figure 2: Voltage measurement schematic

Voltage measurement was one of the most important design aspects of our project. When laid out in our project specifications, we ultimately decided that we wanted the voltage to be measured in DC between $\pm\ 30\ V$ and we wanted that measured value to be within $\pm$10% of any commercially available device that it was put up against. Our final design for voltage measurement can be seen above in Figure 2, which was composed of two separate circuits. Initially, we considered using a processor that was capable of taking differential measurements as we thought this would be the way that our voltage measurements were taken. However, we found a different method where that differential measurement requirement was not needed. This allowed us to switch our processor to the ATmega328p which was cheaper, programmable in the Arduino IDE, and had more open pins which was helpful as we ran out while using the STM32F303. The circuit itself makes use of two separate voltage dividers with a known ratio, 10.1:1. Using this ratio and some simple math, we were able to take two different voltage measurements between our probes and subtract one from the other. The voltage divider also allowed us to get our input voltage into our Op-amps to be $\pm$3V, as we only had a $\pm$3V supply rail available to run our components. The buffer op-amps added before the ADC input are meant to prevent input impedance from our test leads, and thus making our measurements more accurate. For safety reasons, we also added 500 mA fuses just after leads to prevent the ADC from burning up should an unexpected current be placed through the leads. These components were ultimately decided to be simple SMD components due to their price, though they can be replaced by the right person.

3.2 Current Measurement

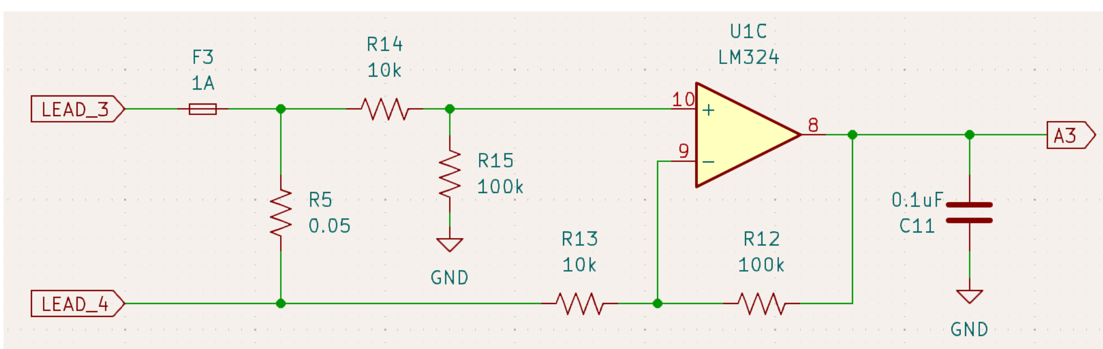

Figure 3: Current Measurement schematic

Current measurement was another critical aspect of our design. During the research phase of the project, it was identified that we want the device to be capable of measuring up to 1 amp of DC current. Due to cost reasons, a dedicated current measurement IC as used in a previous semester of the project, could not be used in this design. Instead, an op-amp current shunt resistor approach was taken which significantly reduces the cost of the current measurement circuit. During the design phase, initially the current was measured using a 1 Ohm shunt resistor, which produced a relatively large voltage drop across the load. This simplified measurement, higher resistance produces a higher measurable voltage, but it also introduces significant power dissipation ($P = I^2 * R$), especially at higher currents. We were planning on the maximum measured current being 1 Amp, meaning we would have 1 Watt of Power going across it. During the Critical Design Review (CDR), we determined that it would be best to use a smaller resistor to ensure safe usage. The wattage rating we would need for the 1 Ohm resistor would be around 2 Watts, considering you want it to be at least double the expected load, in comparison to .1 Watts for the .05 Ohm resistor. Considering the budget, the .05 Ohm resistor made the most sense.

When taking current measurements, we used a shunt resistor and op-amp configuration based on the circuit described at *labprojectsbd.com*. As mentioned above, the shunt resistor we chose, .05 Ohms, produces a small voltage in the mV range. In order to reach a value more easily measured, it was fed into an LM324, which amplified the signal before sending it to an ADC input (A3). After the amplified values were fed into the ADC inputs, they reached the microcontroller, the ATMEGA328P-AU. The current is then calculated. First, the .05 Ohm shunt resistor converts the current into a voltage drop according to Ohm’s Law, for example, a 1 Amp current would produce around 50 mV of voltage. That drop is then amplified by the LM324 op-amps before feeding into the ADC (A3) of the ATMEGA328P-AU, where the current is calculated using the formula below:

$$I = \frac{Vout}{Rshunt * Gain} \quad [1]$$

Where Vout is the amplified voltage, Gain is the gain of the op-amp, and Rshunt is the resistance of the shunt resistor. In the system, there is a 1 Amp fuse that helps to ensure that currents above what can be handled don’t destroy the entire system. When choosing the components for the current measurement, the parts were picked out to be relatively easy to hand-solder.

3.3 Resistance Measurement

An initial idle goal we considered was integrating resistance measurement into our design. We considered 100000 ohms to be a plausible max if we were to get resistance measurement to work properly. Initial research into the subject gave examples of ohmmeters that only had the ability to measure in milliohms. Going further, we consulted the larger Digital Multimeter project as they had a method to measure resistance in the megaohm range. They in turn had used a design they found online repurposed for their needs [12]. This gave us a good starting point, but more needed to be done if we wanted to use this same design in our own project. Several hours were devoted to figuring out how each part of the ohmmeter worked and how we could translate those components into surface mount and making it small enough to fit within our size constraints outlined in 3.5 PCB Design. After working at it for a while, it was realized that resistance measurement required more time, energy, and research than we were able to provide so the idea was ultimately scrapped.

3.4 Display and Display Driver

Unlike last semester, space and cost were major factors in our project design. To accomplish this, a new display needed to be selected instead of using the previous 16x2 character LCD from last semester. During our research, we found the LCD-S401M16KR by Lumex to be a potential solution for our needs. Its single unit cost was $1.67 with the price dropping to $0.8991 when ordering a quantity of 100. One characteristic that needed to be considered with selecting this display was how to drive it. Unique to these tiny displays, they want to be driven by an AC, 1/4 duty, 1/3 bias signal while simultaneously switching an internal mux to control what segments will turn on and off. Originally, this was attempted through just using the GPIO pins on the microcontroller, while this did turn certain segments on and off, it used all of the GPIO pins available and didn’t fully work the way we had intended. This ultimately led to the use of the BU91796FS-ME2 display driver IC. This device could be communicated with over I2C, would handle all of the AC signal, 1/4 duty, 1/3 bias, requirements of the display and only cost $1.1022, putting our total cost at just around $2, one of the cheapest options compared to other displays. Pawning off the control of the display to this driver frees up many of the GPIO pins on the microcontroller while also handling all of the specific characteristics with driving the display. While this solution led to more progress than using just the GPIO pins on the microcontroller, only a few numbers were able to be displayed by the end of the semester. With a bit more time and effort, the correct I2C commands could be determined to properly display all possible numbers correctly on the display.

3.4 Programming

The firmware for this device was written in C++ through the Arduino IDE. The ATmega328 processor used in this device is the same processor used for the Arduino Uno's, combine this with the Arduino IDE and there are many libraries available that can be used in the firmware for the device. Additionally, many artificial intelligence (AIs) have been trained with C++ and Arduino programs, which improves the chances that using an AI to assist with programming will lead to a functional result. That being said, AI was used in an attempt to generate many of the I2C commands for the display driver, while it greatly helped finding a majority of the commands, there were a few commands/segments that were not located and would require more troubleshooting and review of the datasheet to correct. Aside from this, the program functions flawlessly and can be downloaded from our GitHub and programmed onto the device. The device will require a bootloader prior to programming from the Arduino IDE, see our references on flashing the bootloader to the ATmega328 for more thorough instructions and troubleshooting steps. Once the bootloader has been flashed, the Arduino IDE will now recognize the device and the firmware can be uploaded using a USB to UART adapter or a second Arduino. Plans are in place for a future addition of a USB to UART IC being placed on the board along with a USB port for direct programming of the device.

3.5 PCB Design

After completing each subsection of the schematic, we moved onto the PCB design and layout. There were many versions of the PCB created with various dimensions, component layouts, graphics, etc. in an attempt to get everything to fit onto the board in an intuitive way. Eventually, each subsection of the schematic was laid out separately from the board to ensure each section had its most optimal layout based on the components datasheets and where the user might interact with the device. After which, all subsections were combined into the remaining space on the board and any components that were interfering with one another were relocated. One of the primary ideas that we kept in mind when designing the board layout was keeping all components on the top side of the board, aside from the battery holder and power switch, for this first initial prototype, to make debugging and assembly much easier. After quite a lot of layout tweaking, we reached our final design for our initial prototype shown in Figure 4.

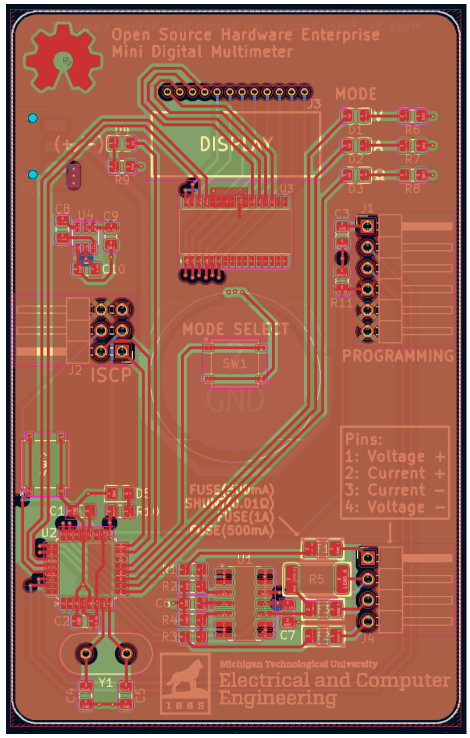

Figure 4: Initial Prototype PCB Design

3.6 Enclosure

The enclosure was designed to allow for easy and quick 3D printing and assembly. It's composed of two different halves, the case and the lid, and assembled using four small phillips head screws. It is printed using PLA filament and is a simple shell to house the device and add some protection to both the device and the user. The working prototype can be seen below in Figure 5, with plans to refine it for future production.

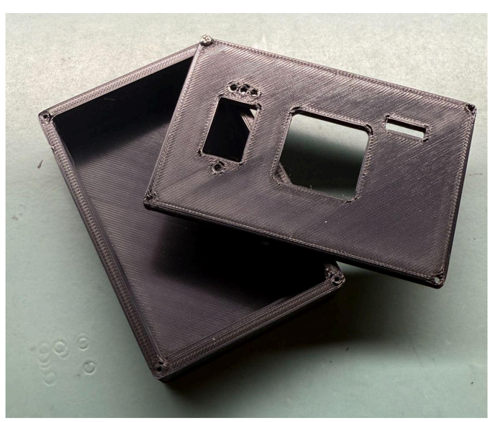

Figure 5: Initial Prototype 3D Printed Case

4. Bill of Materials (BOM)

Table 1: Bill of Materials

Designator	QTY	Component Value	Part Number	Cost
U2	1	MCU	ATMEGA328P-AU	$2.66
U1	1	Op-Amp	LM324DT	$0.25
U4	1	Charge Pump	LM2664M6X/NOPB	$0.92
J3	1	Display	LCD-S401M16KR	$1.67
U3	1	Display Driver	BU91796FS-ME2	$1.85
R5	1	0.01	PA0805FRF870R01L	$0.70
R1,R3	2	9.1 MEG	CRCW08059M10FKEA	$0.10
R2,R4	2	1 MEG	CRCW08051M00FKEA	$0.10
R6, R7, R8, R9	5	1k	CRCW08051K00JNEAC	$0.10
R11	1	10k	CRCW080510K0FKEAC	$0.10
F3	1	1 A	C1F 1	$0.34
F1,F2	2	500 mA	C1Q 500	$0.37
D1,D2,D3,D4,D5	5	Green LED	LTST-C170GKT	$0.10
C1,C2,C3	3	0.1uF	C0805C104K5RACTU	$0.08
C6,C7	2	1uF	CL21B105KBFNNNE	$0.08
C8,C9,C10	3	10uF	CL21A106KOQNNNE	$0.08
C4,C5	2	22pF	CL21C220JBANFNC	$0.10
Y1	1	16MHz	16M20P2/49US	$0.15
SW1,SW2	2	Switch	PTS526SK15SMTR2 LFS	$0.14
SW3	1	Switch	JS102011SAQN	$0.76
J1,J2,J4	1	Pin Header	61301611121	$0.87
BT1	1	Holder	3034	$0.40
BT1	1	Battery	CR2032	$0.33
Screws	4	Screws	#2 x 1/4" Pan Head Sheet Metal Screws	$0.28
PCB	1	JLCPCB	JLCPCB	$2.00
			TOTAL	**$14.53**

5. Tools Used

Table 2: Software Used

Name	Description	Price	Link
GitHub Desktop	Used to share updates and versions.	Free	GitHub.com
KiCAD	Software utilized to design PCB and electrical schematics	Free	KiCAD.com
OnShape	Used to design the enclosure and dial to controls the PCB	Free	OnShape.com
PrusaSlicer	Software used to slice the OnShape file to prepare to be 3D printed.	Free	prusa3d.com
Total Cost		Free	

Table 3: Hardware Used

Name	Part Number	Description	Price	Link
3D-Printer	Original Prusa MINI+	Used to print enclosure for multimeter	$459	prusa3d.com
Soldering Iron	Hakko FX888DX	Used for through hole components and PCB corrections	$118	Amazon.com
Solder Wire	Enersystec		$19.99	Amazon.com
Multimeter	Fluke	The Multimeter was from the OSHE lab to test components, debug test circuits, and accuracy against our Multimeter		
Tweezers	Electronics	Tweezers used to place	$5.99	Amazon.com

	Tweezers	surface mounted components onto the PCB		
Pneumatic Solder Dispenser	METCAL DX-250		$587.00	Metcal.com
Solder Paste	Chip Quik NC191LTA35	Solder paste used to mount surface mount components to the PCB.	$17.95	Digikey.com
Reflow Oven	Manncorp MC301 (Legacy)	The reflow oven was used to melt the solder paste to solder all surface mount components to the circuit board.	Unknown	Unknown
Total Cost			$ May Vary	

6. Assembly Instructions

6.1 PCB Assembly

- Step 1: If using solder paste, apply paste to each SMD pad on the PCB
- Step 2: Place SMD components onto the PCB using a pick and place machine or by hand using tweezers following the reference designators.
- Step 3: Place the board into the reflow oven with the temp probe on top and run the reflow profile for the specific paste you are using.
- Step 4: Verify good solder joints on components, then move onto through hole components in the next step.
- Step 5: Place the through hole components onto the top side of the board, flip over the board and use solder wire and a soldering iron to solder each through hole lead. Refer to the board assembly references section below for hand soldering techniques.

- Step 6: Once all through hole components are placed, cut the excess leads off using a wire cutter.
- Board Assembly References:
    - SMD Reflowing: https://www.youtube.com/watch?v=yZqrrq6VpKY
    - Hand Soldering Techniques: https://www.youtube.com/watch?v=6rmErwU5E-k
    - Hand Soldering Techniques: https://www.youtube.com/watch?v=vAx89WhpZ3k
    - Identifying Bad Solder Joints: https://www.youtube.com/watch?v=qUzoG6O5JI0

6.2 Enclosure Replication

- Step 1: Download the STL files from GitHub and import them into your preferred slicing software.
- Step 2: Adjust print settings such as infill, layer height, etc for the specific speed and/or quality you are looking for out of the part. Usually the standard settings will be fine for these parts.
- Step 3: Print the lid and base for the device.
- Step 4: Place the PCB inside the case. You can use some double sided tape to hold it down if necessary.
- Step 5: Screw the lid of the case using four phillips screws.

7. Characterization Data

Table 4: Characterization Data (OSHE DMM vs Fluke)

Voltage		Current	
OSHE MINI DMM	**Klein Tools**	**OSHE MINI DMM**	**Klein Tools**
4.97 V	4.98 V	00.00 mA	00.00 mA
9.93 V	10.00 V	7.54 mA	8.11 mA
14.95 V	14.96 V	15.13 mA	15.79 mA
20.05 V	20.03 V	28.32 mA	29.11 mA
24.96 V	24.96 V		
29.68 V	29.55 V		

Characterization Testing Notes:

- The voltage measurement test circuit consisted of the mini digital multimeter being connected in parallel with a 30 volt DC power supply and the Klein Tools multimeter.
- The current measurement circuit was three 1k resistors in series which started at nothing being connected, then four resistors, then three resistors, then two resistors, and finally just one resistor.

8. Sponsors/Acknowledgements

- This project is currently self sponsored by the MTU Open Source Hardware Enterprise.

9. References

[1] Github Repository: https://github.com/peterk-mtu/oshe-dmm

[2] "Digital multimeter (DMM) design resources | TI.com," Ti.com, 2025. https://www.ti.com/solution/digital-multimeter-dmm?variantid=20220#block-diagram (accessed Apr. 24, 2025).

[3] "Arduino as ISP and Arduino Bootloaders," Arduino.cc, Jan. 20, 2018. https://docs.arduino.cc/built-in-examples/arduino-isp/ArduinoISP/

[4] Electronoobs, "Handheld Multimeter," electronoobs.com, Dec. 01, 2024. https://electronoobs.com/eng_arduino_tut194.php#google_vignette

[5] IPC, *IPC-2221: Generic Standard on Printed Board Design*, Northbrook, IL: Institute for Interconnecting and Packaging Electronic Circuits, Feb. 1998. [Online]. Available: https://www-eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC_2221.pdf

[6] M. Das, "DC Current measurement using Shunt resistor and Op-Amp Circuit" *Lab Projects BD*, Sep. 05, 2020. https://labprojectsbd.com/2020/05/09/dc-current-measurement-using-shunt-resistor-and-op-amp-circuit/

[7] M. R. Thakur and A. Y. says:, "DC current measurement using shunt resistor," Circuits4you.com, https://circuits4you.com/2016/05/13/dc-current-measurement-arduino/#google_vignette (accessed Dec. 5, 2025).

[8] M. James, "Build your own ohmmeter!," *Programming Electronics Academy*, Sep. 08, 2021. https://www.programmingelectronics.com/ohmmeter/ (accessed Dec. 04, 2025).