AVRpsu

Microcontroller Based Laboratory Power supply

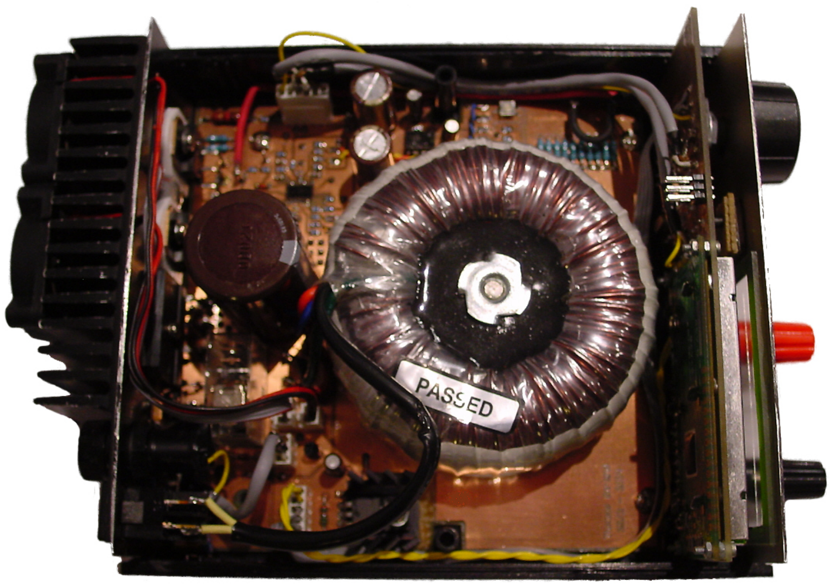
Written with LyX in April - May 2004 by Thomas Strand.

Contents

Chapter 1

Introduction

Any engineer, technician or hobbyist need some form of power supply when designing or experimenting with electronic circuits. The power supply described here will cover a broad range of applications, including battery charging. AVRpsu is a laboratory power supply with a single output of 0-25V, 0-2.5A. Output voltage and current limit are adjustable from the front panel controls or remotely through the RS-232 port, with resolutions of 100mV and 10mA. The design is entirely linear, resulting in lower noise than switched designs. A very compact size is achieved through use of a highly integrated microcontroller, the Atmel ATmega8. It eliminates the need for external devices such as A/D- and D/A-converters, memories, clock circuits or any glue logic. This saves on design time, cost and space.



Chapter 2

Operation

2.1 Overview

AVRpsu may be operated either from the RS-232 port or from the front panel. The front panel consists of an alphanumeric display, two buttons, a rotary encoder and a pair of output terminals. The two buttons are labeled SET and OUTPUT, and are used for selecting which parameter to adjust and switching the output on or off, respectively. A rotary encoder is used for adjusting parameters. This simple scheme makes operation intuitive, eliminating the need for an in-depth study of the manual before using the unit.

The OUTPUT key is very convenient for switching off the power while doing modifications to the project at work.

Pressing and holding either of the buttons has special functions.

For operation from the RS-232 port, see section 2.4.

2.2 Adjusting Parameters

Parameters are adjusted from a set of menus, which are entered by pressing/holding the keys. The menu structure is illustrated in Figure 2.1 and is explained in detail in the following sections.

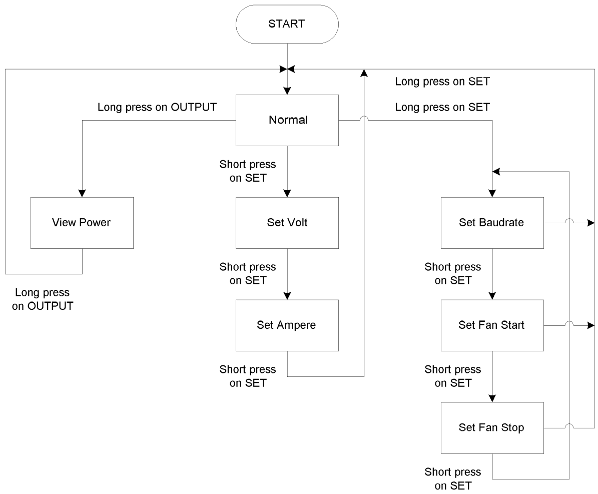


Figure 2.1: AVRpsu menu structure.

2.2.1 Adjusting Voltage and Current

From *Normal* mode, a short press on SET will enter the *Set Volt* mode. Every succeeding short press will advance to the next mode in the list.

Set Volt: Voltage digits are flashing and the voltage setpoint can be adjusted with the rotary encoder.

If no adjustement is made for about 20 seconds, AVRpsu returns to *Normal* mode, leaving parameters in EEPROM unchanged. This is true even if parameters are adjusted from the RS-232 port.

Set Ampere: Ampere digits are flashing and the current limit can be adjusted with the rotary encoder.

If no adjustement is made for about 20 seconds, AVRpsu returns to *Normal* mode, leaving parameters in EEPROM unchanged. This is true even if parameters are adjusted from the RS-232 port.

Normal: Return to *Normal* mode. The voltage and current setpoints are saved to EEPROM, and the cycle is repeated.

2.2.2 Switching the Output On or Off

The output can be toggled on or off with a short press on OUTPUT in any of the modes *Normal*, *Set Volt* or *Set Ampere*. When the output is off, "OFF" appears in the output mode field of the display (see section 2.2.5).

2.2.3 Adjusting Baudrate and Fan trip points

From *Normal* mode, press and hold SET to enter the *Set Baudrate* mode. Every succeeding short press will advance to the next mode in the list.

To exit to *Normal* mode again, press and hold SET. The Baudrate and the Fan trip points are saved to EEPROM.

Set Baudrate: The Baudrate is flashing and can be adjusted with the rotary encoder.

Ten Baudrates are available: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200 bps.

The frame format is fixed to 8 data bits, no parity and one stop bit.

If no adjustement is made for about 20 seconds, AVRpsu returns to *Normal* mode, leaving parameters in EEPROM unchanged. This is true even if parameters are adjusted from the RS-232 port.

Set Fan Start: The Fan Start Temperature is flashing and can be adjusted with the rotary encoder.

It can be adjusted in the following range: $Fan\,Stop + 1 \Longrightarrow 80$

If no adjustement is made for about 20 seconds, AVRpsu returns to *Normal* mode, leaving parameters in EEPROM unchanged. This is true even if parameters are adjusted from the RS-232 port.

Set Fan Stop: The Fan Stop Temperature is flashing and can be adjusted with the rotary encoder.

It can be adjusted in the following range: $30 \Longrightarrow Fan\,Start - 1$

If no adjustement is made for about 20 seconds, AVRpsu returns to *Normal* mode, leaving parameters in EEPROM unchanged. This is true even if parameters are adjusted from the RS-232 port.

Set Baudrate: Return to *Set Baudrate* mode, and the cycle is repeated.

2.2.4 Viewing Output Power and Heatsink Temperature

This mode enables the user to see the power consumption of the load and the measured temperature at the heatsink.

View Power mode is entered by pressing and holding OUTPUT. The power delivered to the load and the heatsink temperature are displayed. Please note the following:

- This mode can only be entered from *Normal* mode.
- No parameters can be adjusted in this mode.
- Short presses on OUTPUT have no effect in this mode, implying the output can not be switched on or off in this mode.

To exit to *Normal* mode, press and hold OUTPUT.

2.2.5 Display

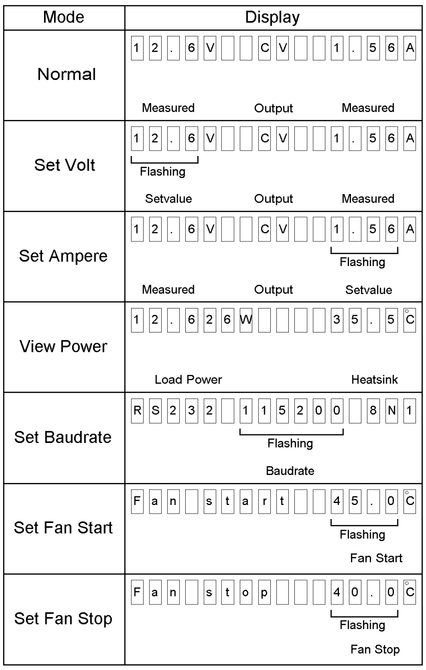


Figure 2.2: Display modes.

The display is a LED backlit, 16 characters $\times$1 line alphanumeric LCD module (Seiko L1671B1J). It has a built-in ASCII translator table, and eight user-definable characters.

The alphanumeric format is more flexible than traditional seven-segment types, allowing all sorts of text and icons to be displayed. This makes menu-based configuration easier to implement, and simplifies adding features.

Different parameters are displayed according to the selected mode, as shown in Figure 2.2.

Flashing parameters can be adjusted by turning the rotary encoder. Adjusting a parameter will cause the flashing to stop for a few seconds, to simplify reading the parameter.

In the modes *Normal*, *Set Volt* and *Set Ampere*, output mode is shown in the center of the display. output mode lets the user know whether the voltage or the current is being regulated (held constant), or if the output has been switched off. Normally, *Constant Voltage* (CV) is the preferred mode and *Constant Current* (CC) means the circuit draws more current than expected. This is, however, not always the case.

When charging NiCd batteries, for example, *Constant Current* is the appropriate mode.

Output Mode is one of the following:

CV	Constant Voltage
CC	Constant Current
OFF	Output has been switched off

2.3 Parameter Storage

All parameters are saved in non-volatile memory (EEPROM) and restored at power-up. This makes it convenient to continue working after the power has been switched off, eliminating the need for re-adjusting parameters. One exception is the output state, which is always off at power-up, to prevent accidental destruction of connected equipment.

Parameters are saved on two occasions:

- When in *Set Ampere* mode and pressing SET for a short time. AVRpsu will return to *Normal* mode and voltage and current limit setpoints are saved to EEPROM.
- When in *Set Baudrate, Set Fan* Start or Set *Fan Stop* mode and pressing and holding SET. AVRpsu will return *to Normal* mode and Baudrate and Fan trip points are saved to EEPROM.

Parameters received through the RS-232 port will not overwrite those stored in EEPROM.

2.4 RS-232 Port

The RS-232 port can be used for remote control of the AVRpsu, including readback of measured values. The port is always activated and will always respond to received commands. The available commands are shown in table 2.1.

Unsupported commands are replied with a "?".

Data values "dd" for different parameters are as follows:

Voltage:	Decimal numbers in the range 0 - 250 representing 0 - 25.0V. This applies for both reading and writing, and illegal values are ignored and replied with a "?".
Ampere:	Decimal numbers in the range 0 - 250 representing 0 - 2.50A. This applies for both reading and writing, and illegal values are ignored and replied with a "?".

Table 2.1: RS-232 port commands

Command	Host writes		Host reads	
	ID	Data	Data	
Set Volt	″V″	dd		13d
Set Ampere	″A″	dd		13d
Read Volt	″v″		dd	
Read Ampere	″a″		dd	
Read Temperature	″t″		dd	
Output on	″O″			13d
Output off	″o″			13d
Read Output State	″s″		dd	

Temperature: Decimal numbers in the range 0 - 255 representing 0 - 127.5 degrees C. The LSB of the data byte represents 0.5 degrees and the remaining seven bits represents 0 - 127 degrees C.

Output State: One of the following characters:

- ′O′: Off
- ′V′: Constant Voltage (CV)
- ′C′: Constant Current (CC)

When a valid command and parameter has been received through the RS-232 port, *Normal* mode will be entered. An icon will appear in the display, at the left side of the output mode field. This indicates one or more parameters has been altered since the last setting from the front panel. The icon will be cleared when a parameter is adjusted from the front panel.

2.5 Firmware Upgrade

The internal firmware of the AVRpsu can be upgraded through the RS-232 port. This is possible due to the boot loader support of the Atmega8, a feature enabling it to reprogram its own program memory.

The program memory (Flash) and EEPROM memories can be read and written through the RS-232 port, but no provisions are made for reading or writing Fuse Bits. This is done in order to prevent the user from rendering AVRpsu unusable by programming a Fuse Bit combination that do not work. One example of this is selecting a clock configuration that does not comply with the actual clock in AVRpsu.

Two requirements must be met to be able to upgrade the firmware:

- Boot Loader Mode must be entered.
- A PC with programming software must be available.

Holding SET while powering up AVRpsu enters Boot Loader Mode. The display reads "Boot Loader Mode".

The programming software must be compliant with Atmel Application Note AVR109. AVRpsu was designed and tested with AVRprog, but other compatible programs are available and should work, though they are not tested.

The firmware can now be upgraded with the programming software. The display will not change during the firmware upgrade. When the upgrade is finished, power-cycle AVRpsu and it is ready for use with the new firmware.

It is not possible to damage the Boot Loader by upgrading the firmware, implying the Boot Loader can not overwrite itself.

Chapter 3

Technical Description

3.1 Overview

AVRpsu is a fairly conventional design, with a few exceptions. The traditional potentiometers for adjusting voltage and current limit are replaced by a microcontroller, which generates two analogue voltages in place of the potentiometers. The microcontroller also takes care of measuring the voltage and current at the output, as well as controlling the relay in the Raw Power Supply, measuring the temperature at the heatsink, and controlling the cooling fans. In addition to all this it provides a RS-232 port, which can be used for remote control of the unit. Figure 3.1 shows the AVRpsu block diagram.

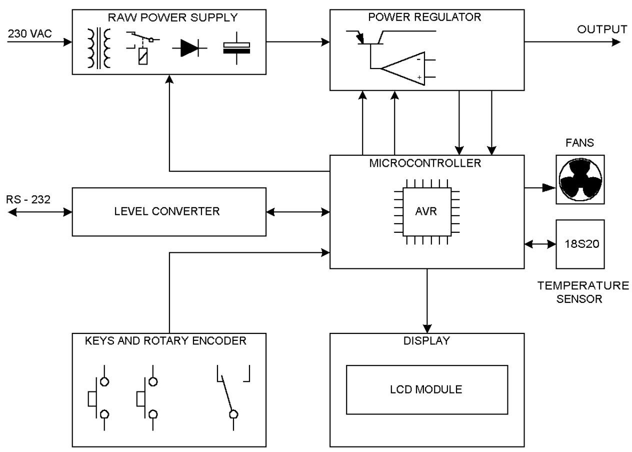


Figure 3.1: AVRpsu block diagram.

3.2 Analogue Circuit

Figure 3.2 shows the schematic for the main board, containing all the analogue circuits. It is followed by a detailed description for each main portion of the circuit.

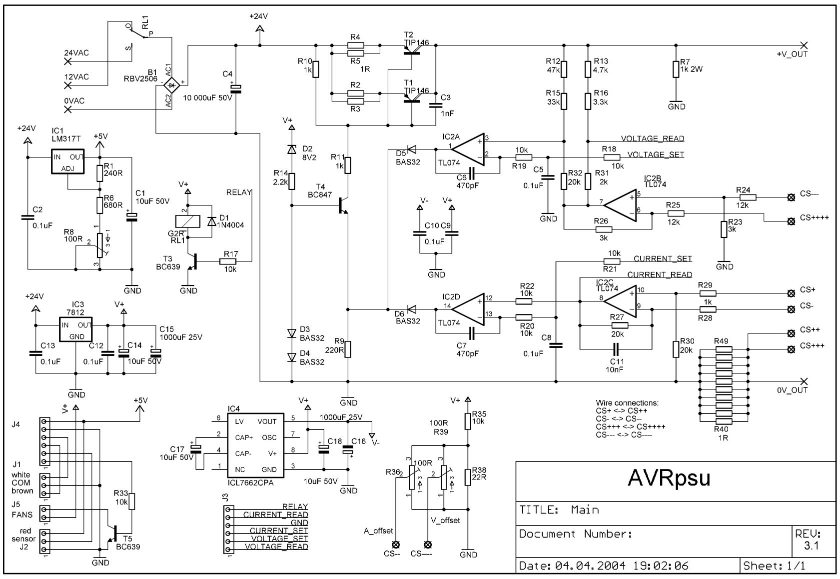


Figure 3.2: The main board schematic reveals a straightforward design.

3.2.1 Raw Power Supply

The Raw Power Supply provides the raw DC voltage to the Power Regulator and consists of a mains transformer (not in schematic), relay (RL1), rectifier (BR1) and reservoir capacitor (C4). This voltage contains a fairly large amount of ripple, but the Power Regulator suppresses this, so there is virtually no ripple voltage at the output. The voltage is divided in two ranges in order to minimize power dissipation in the output transistors. A mains transformer with dual 12V secondaries connected in series provide the two voltage ranges. A relay (RL1) switches between the two secondary voltages, resulting in 12 or 24V AC to the rectifier. The relay is in turn controlled by the microcontroller, operating it according to the measured output voltage.

Power dissipation is limited by the heatsink; the output transistors are capable of dissipating all power that can be generated under any condition. Due to this fact, the relay does not switch from 24V to 12V immediately when the output voltage drops below the threshold, but does so after a delay of a few seconds. On the up-going slope, however, the relay operates immediately, in order to track the setpoint voltage when the user is increasing the voltage. This procedure ensures minimum wear and maximum life span to the relay.

3.2.2 Power Regulator

The power regulator is a classic analogue circuit built from operational amplifiers and transistors. It regulates both voltage and current at the output terminals to limits set by the user. All regulation is done in this circuit; the microcontroller is not part of the regulation loop. The microcontroller only provides the setvalue signals for the regulator. Scaled signals for measuring output voltage and current are also provided by the regulator circuit.

3.2.2.1 Output Stage

The output stage consists of Darlington transistors T1 and T2 with emitter resistors R2 - R4 ensuring equal current distribution. PNP transistors are selected because they enable the voltage drop to be minimized. The transistor pair obtains their base current from the constant current source built from T4, D2, R14, D3, D4 and R9. The current source draws about 3mA of base current for the output transistors, which is sufficient to drive them into full conduction. However, op-amps IC2A and IC2D interfere with the circuit, preventing it from drawing the full 3mA. The op-amps thereby form parts of the voltage and current regulator.

If the output of one or both of these op-amps goes positive, the voltage at T4's emitter will rise, the base current for T1 and T2 will decrease, and the output voltage will decrease. This will happen if the output voltage is higher than the setpoint voltage, or if the output current is higher than the setpoint current.

3.2.2.2 Voltage Regulator

IC2A compares the actual output voltage, VOLTAGE_READ, with the setpoint voltage, VOLTAGE_SET. If the actual voltage rise above the setpoint, the output of IC2A will rise, the Emitter of T4 will rise, and the Base current for the output transistors will decrease, counteracting the initial voltage rise.

VOLTAGE_READ is a DC voltage in the range 0 - 5.12V. It is the output voltage divided by 5, the voltage divider being formed by R12, R15 and R32.

VOLTAGE_SET is a filtered PWM signal generated by the microcontroller.

The AD-converter signal is labeled VOLTAGE_READ in Figure 3.2 and has its own voltage divider formed by R13, R16 and R31. This is done in order to prevent erroneous voltage readings under some conditions. This occurs when the actual voltage is much lower than the setpoint voltage, because the op-amp will not allow large voltage differences between its inputs. The voltage at the non-inverting input of IC2A will then not reflect the output voltage, but will be pulled up by the inverting input of the op-amp.

The lower end of R32 could have been connected to ground, but that would have caused the voltage regulator not to compensate for the voltage drop across the current sense resistor R40 - R49. IC2B ensures that this voltage drop is compensated for by lowering the lower end of R32 to a voltage more negative than ground. This fools the voltage regulator to believe the output voltage is lower than it actually is. This portion of the circuit is borrowed from a design featured in the magazine Allt om Elektronik (Swedish version of Elektor).

3.2.2.3 Current Regulator

IC2D compares the actual output current, CURRENT_READ, with the setpoint current, CURRENT_SET. If the actual current rise above the setpoint, the output of IC2D will rise, the Emitter of T4 will rise, and the Base current for the output transistors will decrease, counteracting the initial current rise.

The actual current, CURRENT_READ, is is simply the voltage across the 0.1Ω 6W current sense resistor (R40 - R49) multiplied by 20. This results in a voltage in the range 0 - 5.12V representing 0 - 2.55A. This signal is also fed to the AD-converter.

The setpoint, CURRENT_SET, is a DC voltage in the range 0 - 5.12V, and is a filtered PWM signal generated by the microcontroller.

3.2.3 Thermal Management

A Dallas 18S20 temperature sensor connected to J2 and two fans connected to J5 provide for thermal management. The temperature sensor is fixed to the heatsink close to the output transistor pair. It has a 1-wire digital interface, occupying only one pin on the microcontroller. The heatsink temperature is monitored continuously and the cooling fans are controlled according to trip-points defined by the user.

If the heatsink should reach unacceptable temperatures, an over-temperature mode is entered. In this mode the output is switched off, the fans are started and a message scrolls through the display telling what has happened. When the unit has cooled down, normal operating will be resumed. The output will remain in the off state, however.

3.2.4 Auxiliary Circuits

The rest of the main board schematics consist mainly of voltage regulators and interconnections.

IC1 is an adjustable voltage regulator, LM317, which provides the +5.12V for the digital circuits. This voltage is adjusted by the potentiometer R8, which thereby set the gain of both voltage and current regulator loops.

IC3 is a fixed voltage regulator, 7812, which provides +12V for the relay, fans, op-amps and bias for the output transistors.

IC4 is a voltage converter, which converts the +12V to -12V. Used for the op-amps.

T3 is the relay driver and T5 is the driver for the fans.

R36 and R39 are the offset adjustment potentiometers for the current and voltage regulators, respectively.

3.3 Digital Circuit

Figure 3.3 shows the schematic for the display board, containing all digital circuitry.

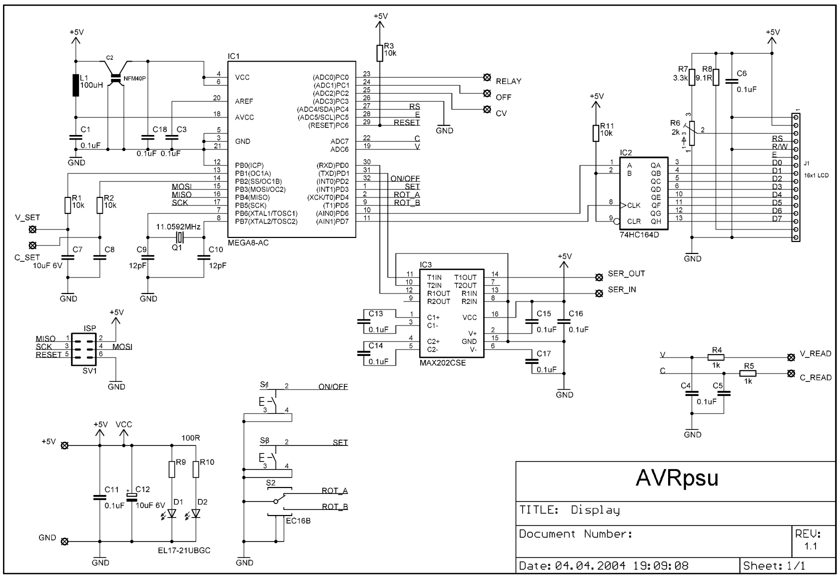


Figure 3.3: Display board schematics. Simplicity seems to be the rule here.

R6 is the contrast adjustment potentiometer for the LCD module.

D1 and D2 are LEDs for illuminating the front panel (not implemented on the prototype).

3.3.1 Microcontroller

IC1 is an Atmel ATmega8 and is the heart of the system, controlling it all. Thanks to all its integrated peripherals, very few external components are needed.

Supply decoupling for the microcontroller is provided for by C2 and C18. C2 is a Murata filter designed to filter supply lines of digital circuits. It prevents switching noise from the microcontroller from entering any analogue circuits.

The ISP (In-System Programming) connector was used during development, for loading the program into the microcontroller. This is necessary the first time, as no Boot Loader is present when the device is new. The ISP is also necessary for programming Fuses and for experimenting with the Boot Loader.

3.3.1.1 AD-Converter

The internal 10-bit AD-converter has its own supply pin, AVCC, which is connected to the +5V (actually 5.12V) through a LC-filter consisting of L1 and C1.

This arrangement assures clean power to the sensitive circuitry of the AD-converter. The AD-converter is configured to use the AVCC as its reference voltage. This implies a 5.12V input at the active analogue input will result in a full-scale result.

The AD-converter measures the voltage and current at the output, and displays the values on the display. The signals V_READ and C_READ (see Figure 3.3) are fed to the AD-converter through separate low-pass filters formed by R4, C4, R5 and C5. Only 8 bits resolution is used in this design, using the eight most significant bits from the AD-conversion result. The next significant bit is used for rounding, however. If the next significant bit is set, the result is rounded up, otherwise not. This method ensures correct rounding of the result, but the logic behind it can be somewhat difficult to realize at first.

Consider the following example: An AD-converter with a range of 0 - 100 is available but only the two most significant digits are used in the application. The range will then be 0 - 10, ignoring the least significant (rightmost) digit. The result will then be 01 for all "real" results in the range 10 - 19. This means 1.9 is rounded down to 1, which is clearly wrong. Now, if the last digit is taken into account this can be avoided. Rounding the result up for all values of the last digit equal to or greater than 5, will correct the result. The same principle is employed in the AD-conversion routine in AVRpsu.

3.3.1.2 PWM

Timer1 is used in PWM mode for generating the setvalue signals VOLTAGE_SET and CURRENT_SET (see Figure 3.2). This timer has two PWM channels and these signals are fed to the power regulator through separate RC-filters, resulting in two DC voltages. These voltages are proportional to the values in their respective Output Compare Registers, OCR1A and OCR1B. This functionality realizes two DAC's with the lowest possible pin usage and component count. Since the signals are generated entirely in hardware, program flow is only affected when altering the Output Compare Register values.

One drawback is slow response in change of settings, a result of the time constant in the RC-filters.

3.3.1.3 RS-232 Port

The internal USART together with the external level converter (see Section 3.3.2.4) provides a RS-232 port, enabling settings to be adjusted from a computer, and measurements to be fed to a computer. It is used in the asynchronous mode, as per the RS-232 standard.

The microcontroller's own program can also be updated through the serial port, see section 2.5. This enables the advanced user to add features to the unit whenever desired.

3.3.2 Support Circuits

3.3.2.1 Display

The display is a Seiko 16X1 LCD module with LED backlight and built-in controller (Samsung KS0066). The alphanumeric format gives a flexible means of displaying information, not limiting the displayable information to just voltage and current but also serial port parameters, temperature etc. LCD modules like this also cost less than a load of LED digits and a driver circuit.

The display module has an 8-bit parallel interface that is connected to the microcontroller via a 74HC164 shift register (IC2) to save port pins. The Read/Write signal is permanently connected to ground, as no reading from the display is necessary. It is common practice to read the display's Busy Flag after each command, to make sure the module is ready before sending any data or command. This is, however, not entirely necessary as long as the timing requirements are met with good margin.

3.3.2.2 Keys

OUTPUT (S1) and SET (S3) are connected directly to the microcontroller and enable the user to switch the output on and off as well as entering menus for adjusting parameters. The microcontroller has selectable pull-up resistors for each pin, eliminating the need for external resistors.

3.3.2.3 Rotary Encoder

The Rotary encoder (S2) is connected directly to the microcontroller. Again, internal pull-up resistors eliminate the need for external resistors. The encoder has two mechanical switches, which give two electrical signals. These two signals are 90 degrees offset, providing direction information. This results in 4 electrical states, which are sequenced when rotating, the order of sequencing determining the direction of rotation. These states are stored in a 4-byte table in program memory and is used for decoding the rotary encoder. One 4-state cycle corresponds to two mechanical indents and $\frac{1}{12}$ revolution on the encoder.

Initially the encoder pins are read and the index to the corresponding table element is stored in a global variable called *rot_prev*. Next time around, the pins are read again, the index is found and compared to the previously stored one. If they match, no rotation has occurred, and the function returns 0. If they are not equal, two *if* statements determine which direction the encoder has rotated, and the function returns -1 or 1.

Not all 4 transactions between states result in a decoded rotation step because that would result in 2 digital steps per mechanical indent. This is not desirable; so only two transactions per direction are selected to result in decoded rotation. This way the increment/decrement of the parameter matches the mechanical indents of the rotary encoder.

A simpler method is to wait for an edge on one of the signals, and then check the level at the other signal to determine the direction. This has one major drawback, however: The program will hang in that function until the edge is detected. That is not at all acceptable in all applications.

3.3.2.4 Level Converter

IC3 is a Maxim MAX202 and provides the level conversion for the RS-232 port. This is an upgrade from the well-known MAX232, requiring just 100nF capacitors for the charge pump voltage converter. It converts the 0-5V signals at the microcontroller pins to $\pm$12V at the RS-232 port, in both directions.

Chapter 4

Calibration

Calibrating AVRpsu is a simple, but nevertheless important procedure. There are three potentiometers to adjust, which is done in the following manner:

- Connect a digital multimeter, V range, to +5V and ground at J4. Adjust R8 for a reading of 5.12V. Turn R36 and R38 fully clockwise.
- Set AVRpsu to 0.1V and 0.05A. Connect a digital multimeter, mV range, to the output terminals. Adjust R39 for a reading of 100.0mV.
- Set AVRpsu to 1.0V and 0.01A. Connect a digital multimeter, mA range, to the output terminals. Adjust R36 for a reading of 10.00mA.
- Set AVRpsu to 19.9V and 0.05A. Check that the output voltage really is 19.9V, and readjust R8 if necessary.

Chapter 5

Performance

This chapter contains measurements made on the AVRpsu prototype, with comments. All measurements were made with an Agilent 54622D oscilloscope, with some supplementing measurements made with a Kenwood CS-5130. Most measurements were made in a noisy lab, as can be seen on Figure 5.1. This is the resulting scope screen with the probe tip connected to the ground clip. AVRpsu was set to 25.0V and 2.50A.

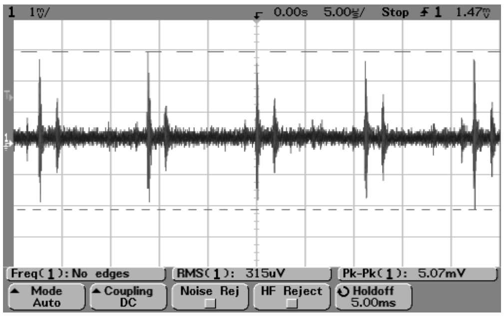


Figure 5.1: Background noise.

All measurements involving low voltages show this noise superimposed on the output voltage. It is important to realize that this noise does not arise in the AVRpsu, but rather in the environment where the measurements were made.

5.1 Summary

Noise:	1.5mV$_{\text{peak-peak}}$
Ripple voltage:	4.8mV$_{\text{peak-peak}}^{\text{max}}$
Ripple current:	13.6mA$_{\text{peak-peak}}^{\text{max}}$
Output resistance:	12.5mΩ
Transient response:	35μs

5.2 Noise

AVRpsu settings: 25.0V 2.50A

Ideally, the output voltage is completely clean under all conditions. In the real world, however, noise and ripple are superimposed on the wanted voltage. Noise is measured with no load connected.

Figure 5.2 shows the noise on the output voltage with no load.

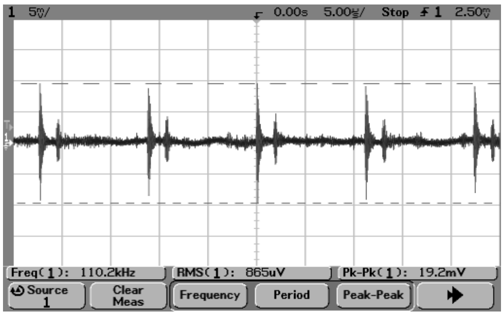


Figure 5.2: Output voltage with noise.

The output voltage resembles Figure 5.1, implying that background noise is very present. If the background noise on Figure 5.1 is subtracted, it can be seen that the AVRpsu noise level is very low.

This measurement was repeated in a different lab, with a Kenwood CS-5130 analogue scope. That revealed about $1.5mV_{pk\text{-}pk}$ of noise and no spikes around 90kHz.

5.3 Ripple

Ripple is variations in the output voltage and current and is caused by the varying voltage across the reservoir capacitor. These relatively large variations arise from the fact that the capacitor is charged with pulses from the rectifier.

5.3.1 Ripple Voltage

AVRpsu settings: 25.0V 2.50A

Ripple Voltage is measured in Constant Voltage mode, with a load resistor resulting in a current of 2.49A. Figure 5.3 shows the output voltage under these conditions.

Noise in the lab is again suspected to be significant. A measurement made in a different lab with a Kenwood CS-5130 scope indicated a ripple voltage of 4.8mV$_{\text{pk-pk}}$ with very little high-frequency components. It matched the darkest portion of the trace in Figure 5.3 very well.

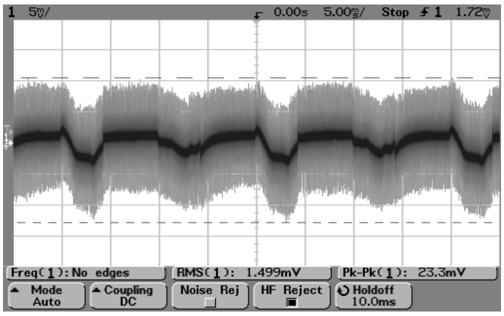


Figure 5.3: Ripple voltage.

5.3.2 Ripple Current

AVRpsu settings: 25.0V 2.50A

Ripple Current is measured as a voltage across a load resistor and calculated using the following formula: $I_{ripple} = \frac{\Delta V}{R_{load}}$

The measurement is performed in Constant Current mode, with a load resistor resulting in a voltage of 24.9V.

According to Figure 5.4, the ripple current is 2.48mA RMS. That is less than 0.1% of the total current, a totally acceptable figure.

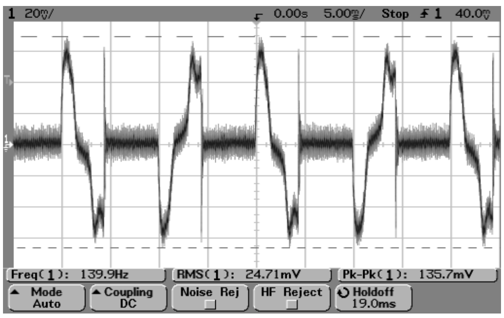


Figure 5.4: Ripple current.

5.4 Output Switching

The output voltage does not change instantaneously when pressing the OUTPUT key. This switching takes time, and the voltage changes in an exponential slope. The slow response is a result of the method used for switching the output off: writing zero to the PWM registers. The delay comes from the low-pass RC-filters for the PWM signals. This may cause problems when working on digital circuits, as they tend to dislike slowly rising power supplies.

5.4.1 From OFF to ON

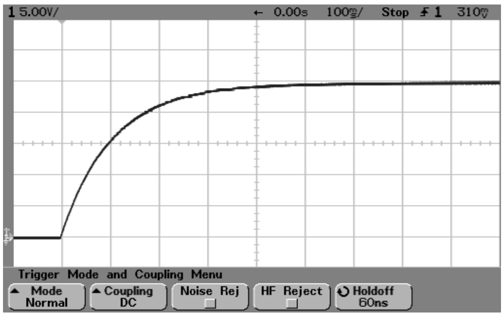


Figure 5.5: Switching from off to on.

AVRpsu settings: 25.0V 2.50A

Figure 5.5 shows the output voltage slope when switching from off to on. The measurement is made with no load.

The slope is slow, but there is no overshoot or any other misbehaviour at all.

5.4.2 From ON to OFF

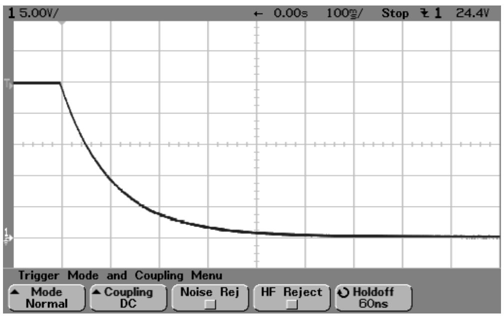


Figure 5.6: Switching from on to off.

AVRpsu settings: 25.0V 2.50A

Figure 5.6 shows the output voltage slope when switching from on to off. The measurement is made with no load.

Again, slow but orderly.

5.5 Output Resistance

AVRpsu settings: 25.0V 2.50A

Output Resistance is defined as the virtual resistance in series with the output of the power supply. This can be seen by the fact that the voltage is not constant for different load currents.

The output resistance is measured using the method illustrated in Figure 5.7. A load resistor is switched on and off by a function generator and a power FET, causing the load to be stepped from zero to a known current at a fixed frequency. AVRpsu is in Constant Voltage mode throughout the measurement.

The output resistance is calculated using the following formula:

$$R_{out} = \frac{\Delta V}{\Delta I}$$

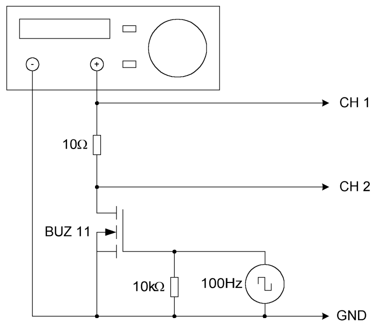


Figure 5.7: Measuring output resistance.

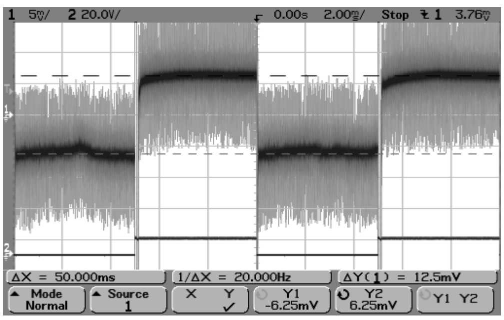


Figure 5.8: Output resistance.

The settings and load resistor result in a current of 1.00A, and Figure 5.8 shows a differential voltage of 12.5mV. This equates to an output resistance of 12.5mΩ, which is a bit on the high side.

5.6 Transient Response

Transient Response is defined as the time the power supply needs to get back on track after a change in load.

5.6.1 Constant Voltage

AVRpsu settings: 10.0V 2.50A

The measurement setup is the same as for output resistance measurement, see Figure 5.7. The only difference is the oscilloscope settings regarding timebase and vertical amplifier.

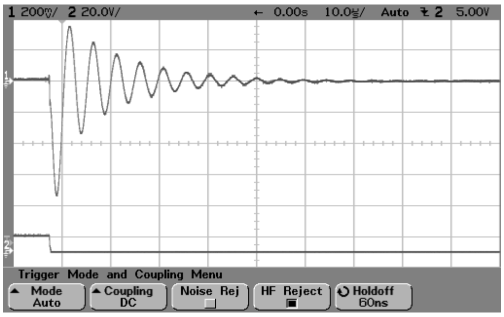


Figure 5.9: Transient response.

Figure 5.9 shows the measured response, Channel 1 trace is the output voltage, AC coupled. The negative edge on the Channel 2 trace indicates the load being switched on. The load current then steps from zero to 1.00A. The Transient Response time is about 35μs.

5.6.2 Constant Current

AVRpsu settings: 10.0V 0.75A

In Constant Current mode, one resistor is permanently connected to the output, while another is switched on and off with a power FET, see Figure 5.10. This way, the output current is changed from one level to another at a fixed frequency. AVRpsu is in Constant Current mode during both these phases.

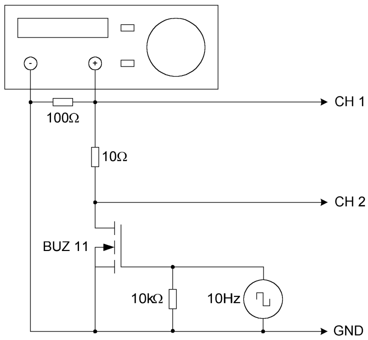


Figure 5.10: Measuring transient response in Constant Current mode.

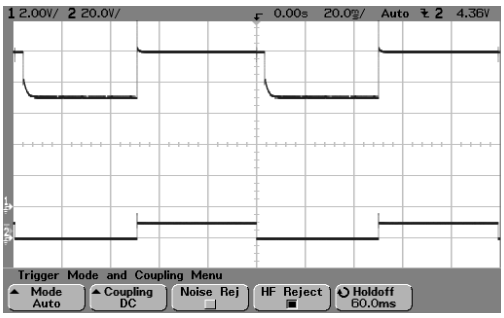


Figure 5.11: Transient response overview.

Figure 5.11 gives an overview of what is going on. Channel 2 represents the extra load resistor being switched on and off (low is on), while channel 1 shows the output voltage change, keeping the total current constant.

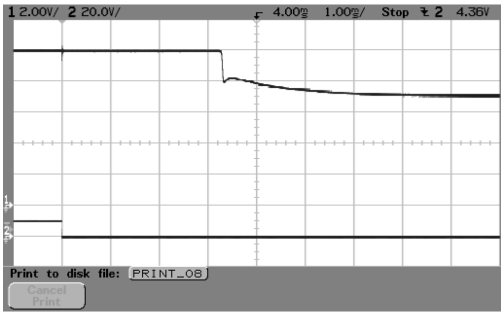


Figure 5.12: Transient response negative edge detail.

Figure 5.12 shows the details of the output slope when the current changes from a low level to a high level. This is the most interesting case, as it represents AVRpsu preventing the load current from growing larger than the user-defined level.

AVRpsu's current regulator is slow. When the load resistance suddenly decreases, AVRpsu does not do anything for about 3.3ms, and then it uses about 3ms to get back on track.

Appendix A

Circuit Board Layouts

A.1 Main Board

A.2 Display Board

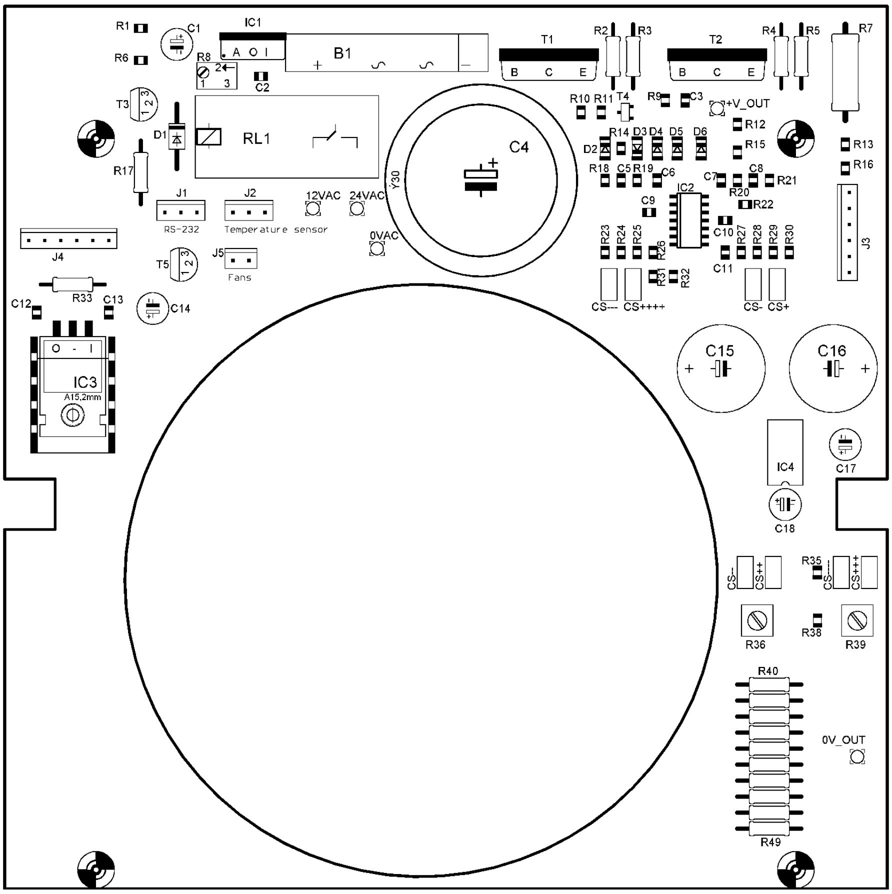


Figure A.1: Main board component placement.

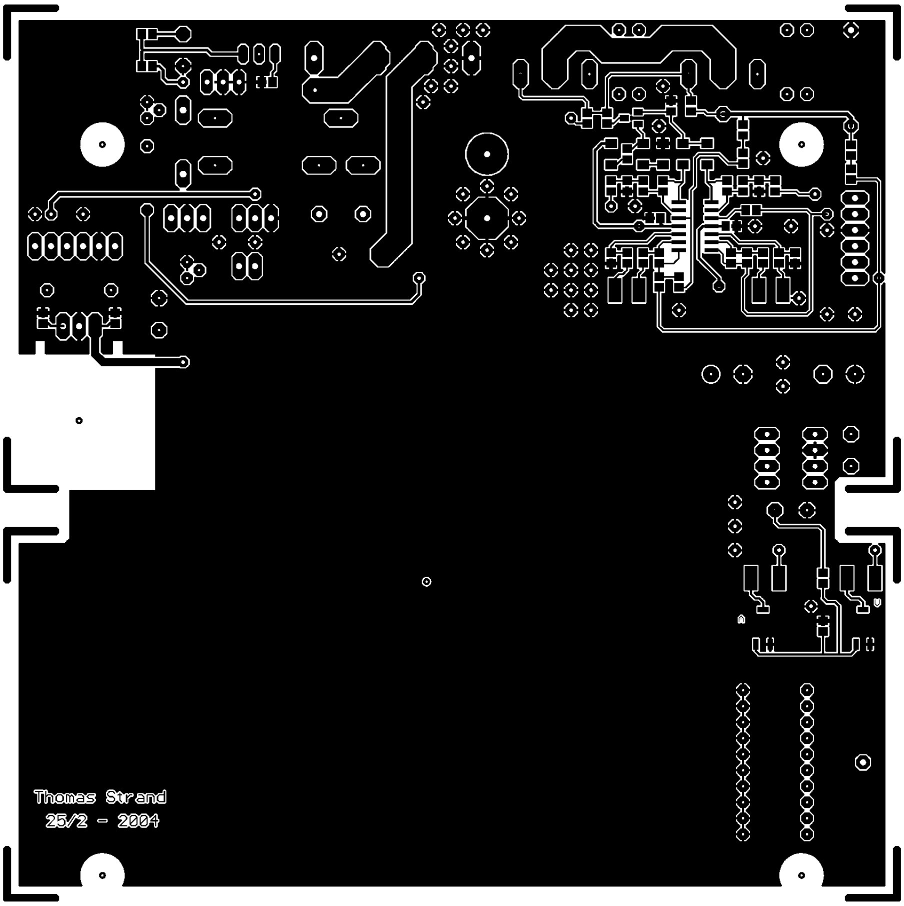


Figure A.2: Main board top layer.

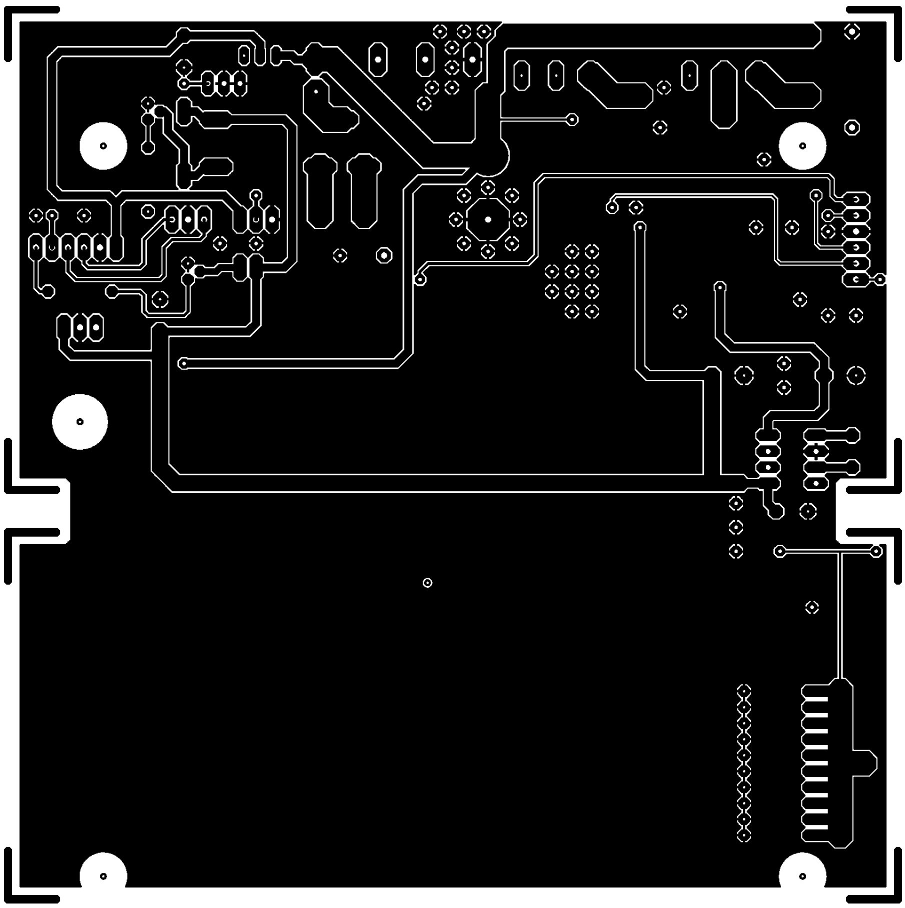


Figure A.3: Main board bottom layer.

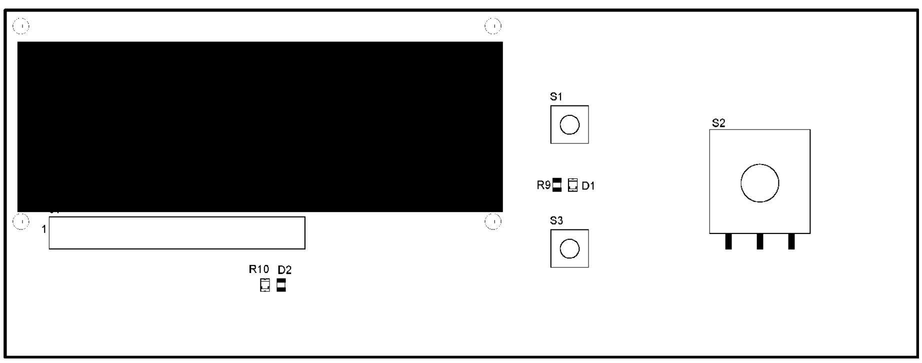


Figure A.4: Display board top layer component placement. Front view.

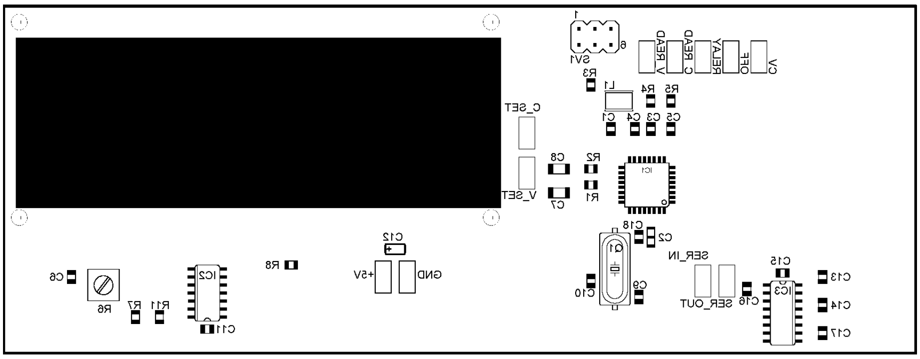


Figure A.5: Display board bottom layer component placement. Front view.

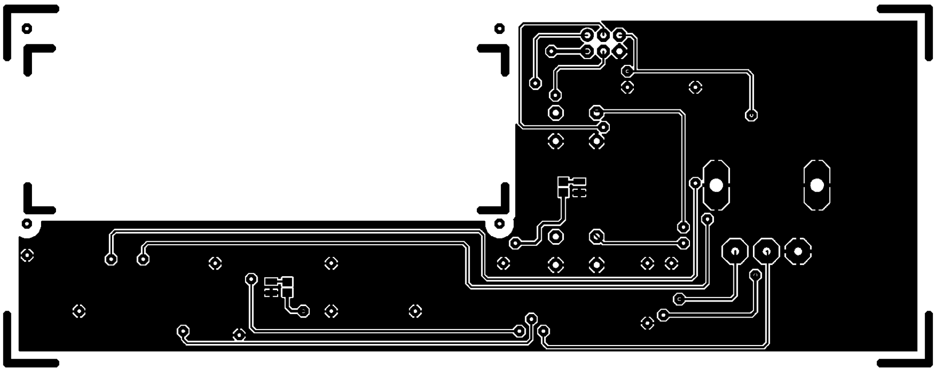


Figure A.6: Display board top layer. Front view.

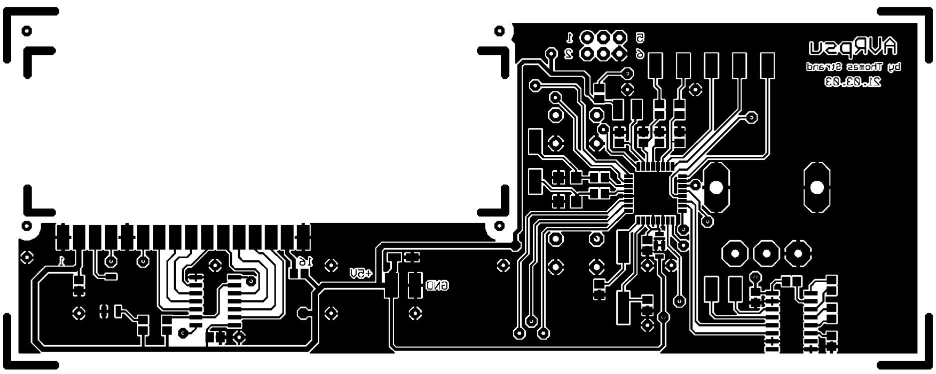


Figure A.7: Display board bottom layer. Front view.