EE 344: Electronics Design Lab

OpenBCI based EEG Acquisition System

Group: TUES-18
Ankith R (200070006)
Anubhav Bhatla (200070008)
Nithish Kumar Reddy (20D070072)

Department of Electrical Engineering
Indian Institute of Technology, Bombay

Contents

Introduction

EEG is a non-invasive method of capturing brain signals. The device consists of two parts: recording electrodes and data-capturing electronics. The electrodes are placed on the subject's head to record different spatial locations on the brain. Each electrode corresponds to a single of data. For example, the OpenBCI Cyton board provides access to 8 channels of data. The Daisy expansion board offers support for an additional eight channels. The goal is to scale the existing design to accommodate 24 channels for better spatial resolution. The design will be robust to ambient noise (including the 50 Hz power supply interference) and has the potential to be used for a wide variety of applications, including medical diagnosis and brain-computer interfacing. The deliverables of the project will include the following,

1. A custom-designed PCB: The PCB will support 24 EEG channels, provide a Wi-Fi module for communicating with a laptop/computer/phone for real-time streaming, a micro-SD card for local storage, and accelerometer support for removing noise due to head motion

2. An EEG headset with electrodes placed at spatial locations recommended by the 10-20 international standard

3. A laptop/phone application to view the data in real-time

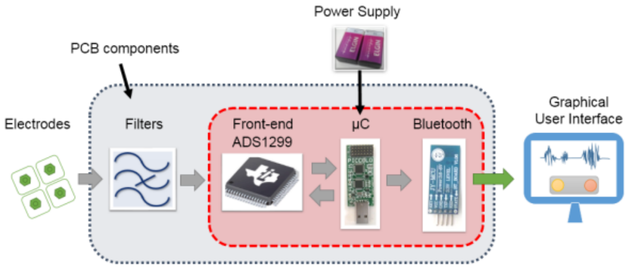


Figure 1: Existing OpenBCI design

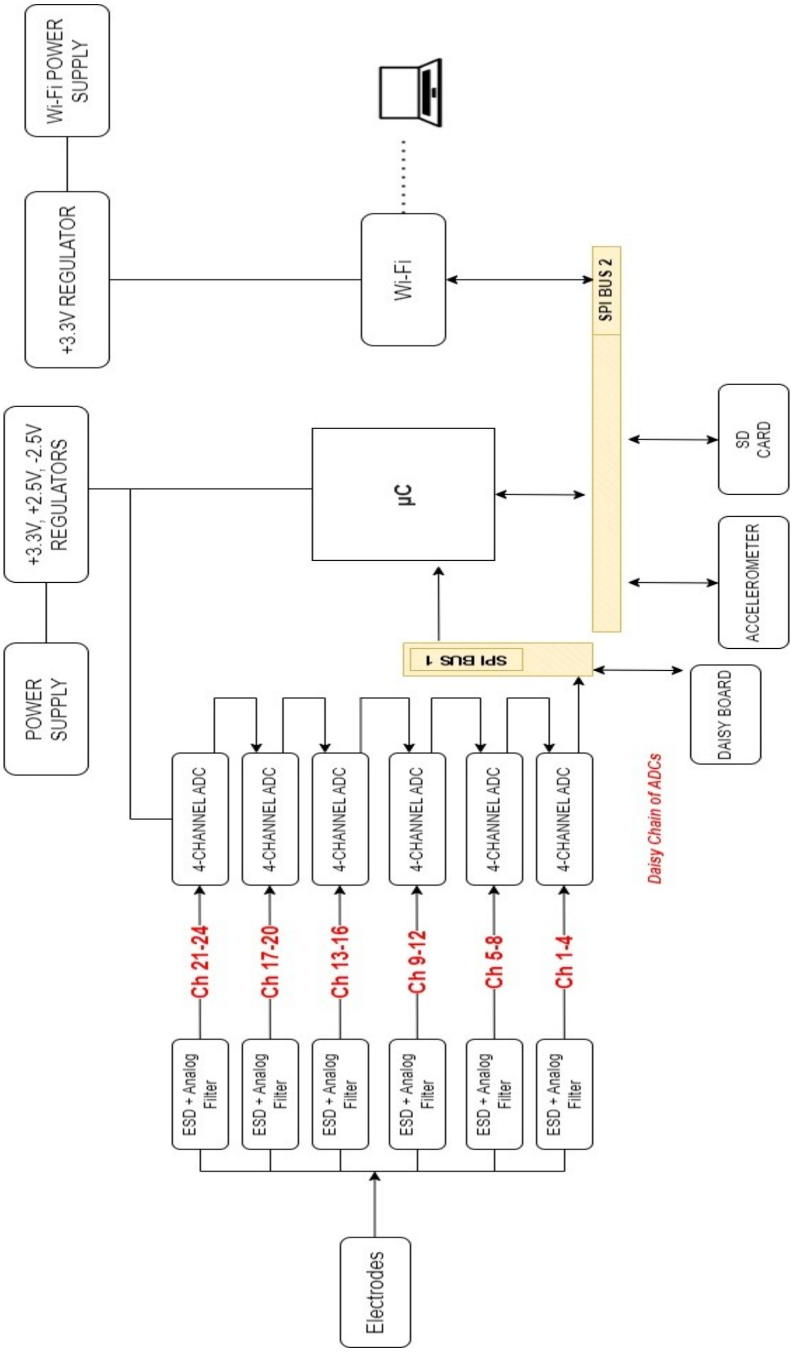


Figure 2: Design Block Diagram

Design Description

The OpenBCI Cyton Board is an 8-channel biosensing board. The existing setup takes in 8 analog inputs from the electrodes and passes them through an 8-channel A/D Converter. The digital output is sent to the microcontroller after which it is passed onto an RFduino Bluetooth Transmitter which transmits the data to a nearby device for viewing.

While scaling up the existing design to meet our requirements, there are multiple factors that we have taken into consideration. Firstly, our targeted 24-channel design would require 6 A/D converters to process the data provided by the electrodes. These devices will peripherally interface with the microcontroller, and therefore, our choice of the microcontroller will have to reflect this additional interfacing. Lastly, limitations on data rate across Bluetooth would impair the performance of our device, especially considering the larger amount of data our module targets to transmit as compared to the original design. To this end, we will look to transmit our data over Wi-Fi for better performance.

A high-level overview of our design is depicted in Figure 2. The subsections that follow give a brief overview of the role and functionality of each module and mention any changes from the existing design that we look to implement.

Electrodes

The EEG electrodes form the analog front end of the EEG processing pipeline and are of different types, including active, passive, dry, and sponge. We will be using the dry spikey and flat electrodes as these have very little preparation time. The electrodes will be mounted on a custom 3D-printed headset, adhering to the 10-20 international standard for electrode placement.

Due to an increased electrode count, specific brain regions can be targeted to study functions like motor function, sensation, and memory. We will specifically target the frontal lobe's motor and visual cortex.

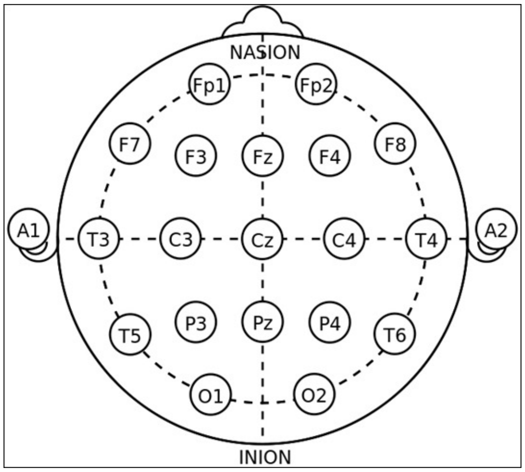


Figure 3: Electrode locations for EEG recording

A/D Converters

The A/D converters are a vital part of the analog front end. In particular, we will use the ADS1194, a low-power, 4-channel, 16-bit analog-to-digital converter for biopotential measurements. One of the major requirements for the A/D converter was the inclusion of amplifiers as the EEG signals are of the order of $10\mu V - 100\mu V$, so we have chosen the ADS1194 which has in-built programmable gain amplifiers.

After being passed through an ESD protection unit, electrode signals from 4 EEG channels are fed into the A/D converter that consists of four simultaneous samplings followed by delta-sigma ($\Delta\Sigma$) converters.

Since the design requires 24 electrodes, 6 A/D converter chips will be on the final PCB. The A/D converters will be connected in the Daisy-chain configuration to ensure synchronisation between them. The output of all the converters is multiplexed into a single stream sent to the microcontroller over a serial interconnect (SPI). The data output sent on the `DOUT` pin consists of 24 control bits followed by 24 sets of 16-bit quantized EEG data.

Microcontroller

The microcontroller at the heart of the system is responsible for gathering data from the A/D Converters and the accelerometer, processing it and passing the final values to the Wi-Fi and SD Card modules for transmission and storage, respectively. The original design for the Cyton board uses a 28-pin PIC32MX250F128B micro-controller, which comes with two I^{2}S/SPI modules for codec and serial communication, up to 13 channel 10-bit A/D Converters, and up to 19 I/O pins. Since the microcontroller will now be interfacing with 6 A/D Converters, the SD Card reader, and the Wi-Fi Module, we will be using the PC32MX250F128D micro-controller, which has a larger number of I/O pins (31), which can be reconfigured as chip select lines for the additional A/D Converters.

The microcontroller will be programmed using a chipKIT UDB32-MX2-DIP bootloader, with code written in the Arduino software library.

Power Supply

The components require a Digital Supply Voltage (`DVDD`) of $+3.3V$, an Analog Supply Voltage (`AVDD`) of $+2.5V$ and an Analog Ground Voltage (`AVSS`) of $-2.5V$. AP2112K-3.3TRG1 is a fixed LDO Voltage Regulator used to generate the `DVDD` signal using the input `RAW` signal. This `DVDD` signal is fed into the LM2664 Switched Capacitor Voltage Converter, which generates the `-RAW` signal. This `-RAW` signal is further passed through the TPS72325 Negative-output Linear Regulator to generate the `AVSS` signal. The `AVDD` signal is generated by passing the `DVDD` signal through the AP2112K-2.5TRG1 Voltage Regulator.

Since the Wi-Fi has high current requirements for the Digital supply regulator, we will be using a separate Voltage Regulator (AP2112K-3.3TRG1) to generate the `DVDD` signal using the input `RAW` signal coming from another battery.

The main consideration while choosing the regulators was the required output voltage and the current rating. The Wi-Fi module requires a large amount of current while transmitting data, so we choose the $600mA$-rated AP2112K-3.3TRG1.
We will use two 3-6V DC Batteries, one as the power source for the board and another for the Wi-Fi.

Accelerometer

The LIS3DHHTR is a three-axis accelerometer with digital I^2C/SPI serial interface standard output, capable of 16-bit data output. The purpose of the accelerometer is to remove artifacts in the EEG signal due to head movements. The data returned by the accelerometer is a good baseline to reconstruct what happened in the user's recording session.

The accelerometer can also be used as a marker for different phases of experimentation. Without the accelerometer, one would have to reset the data logging software. With the accelerometer, one can simply tap the board a few times and the created artifact would be easy to observe in the accelerometer's data stream.

SD Card

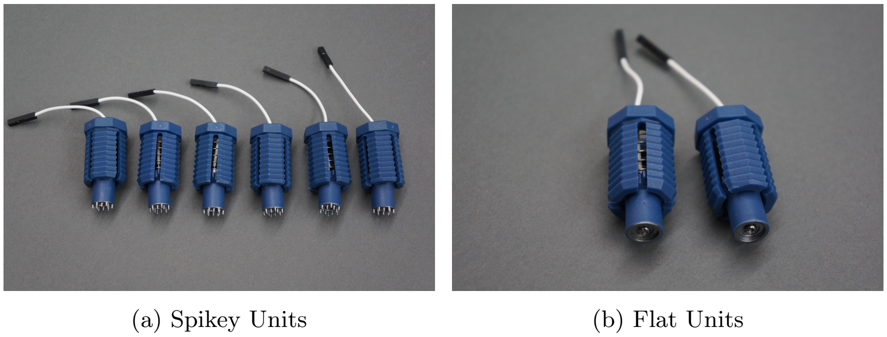
SD card is a necessary provision for logging data to local storage. This is useful in sleep study applications or when it is difficult to make wired connections to the PC. We will be using a Suntech ST-TF-003A SD card holder for the design. The data saved to the SD card is sampled at 250 Hz. This amounts to 3 MB of data per minute and hence, a high-speed SD card with large storage will be used (8 GB, 16 GB, or 32 GB). Data from the A/D converter will be sent to the SD card over an SPI bus.

Wi-Fi Module

Unlike the original design, a BLE module will not be included because of the requirement for frequent firmware updates. Another disadvantage of BLE is that the hardware must support the BLE protocol, which requires data packetization. On the other hand, Wi-Fi is based on a stream protocol which is easier to implement. OpenBCI provides a Wi-Fi shield in addition to the Cyton board, which has been known to suffer from packet losses and cyclical noise. We aim to overcome these defects by providing a reliable data transfer interface using an ESP-12S: ESP8266 Wi-Fi Module, which is Arduino compatible.

Headwear

The Ultracortex Mark IV is a device developed by the OpenBCI company, which allows users to measure and record brain activity (EEG). The following are the main components of the headset: Cables, Spikey units, Flat units, Comfort units, and Ear Clips. Unlike the original design, we have to use 8 more spikey units, i.e., 22 spikey units in total and 2 flat units. In case we also connect the Daisy module, we can use a total of 32 dry electrodes mounted on the headwear frame.



(a) Spikey Units

(b) Flat Units

Figure 4: Electrodes used for EEG recording

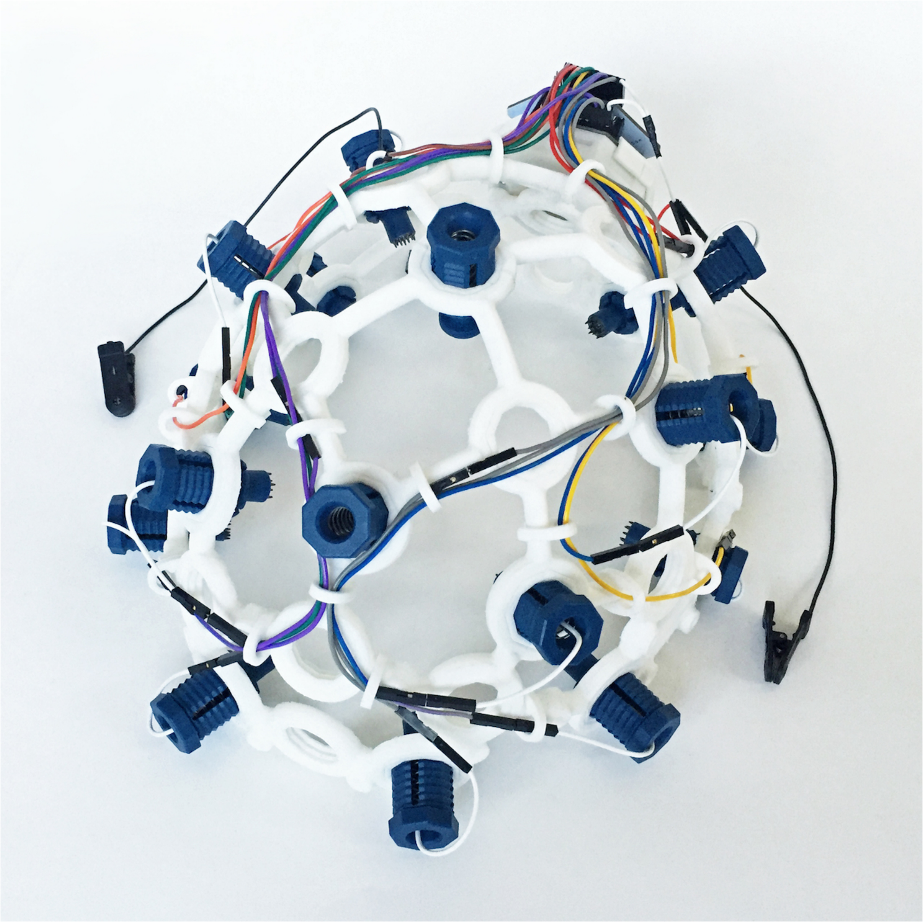


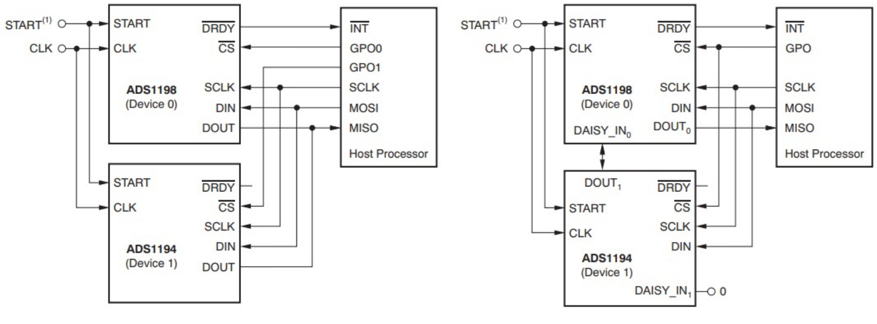
Figure 5: Assembled headwear with electrodes

Principle of Operation

ADC Topology

The original OpenBCI design had each of the ADCs communicate to the microcontroller through a single SPI bus. This was achieved by having each device send its data on the bus by taking turns. In order to implement this, each device had a designated chip select signal, and the microcontroller would selectively turn on a single chip select as needed. However, this scheme has drawbacks in the form of channel data being asynchronous. Furthermore, preliminary analysis of the firmware code suggests that it may be possible to implement a design wherein all the ADCs may be simultaneously selected to transmit their data.

We will look to assemble our ADC subsystem in a **Daisy chain** topology. The Daisy chain is an efficient means to assemble a large number of ADCs in order to obtain high channel counts. In a daisy chain, a single chip select from the microcontroller is shared by all the participating devices. The output of each ADC (D_{OUT}) is connected to the $DAISY_{IN}$ of the succeeding ADC. The result of this chain is that the final ADC in the chain outputs a stream of data corresponding to the outputs of all the preceding ADCs as well as its own data. Therefore, we only require this single ADC to communicate with the microcontroller through an SPI bus.



a) Standard Configuration

b) Daisy-Chain Configuration

Figure 6: Comparison between standard operation and Daisy operation

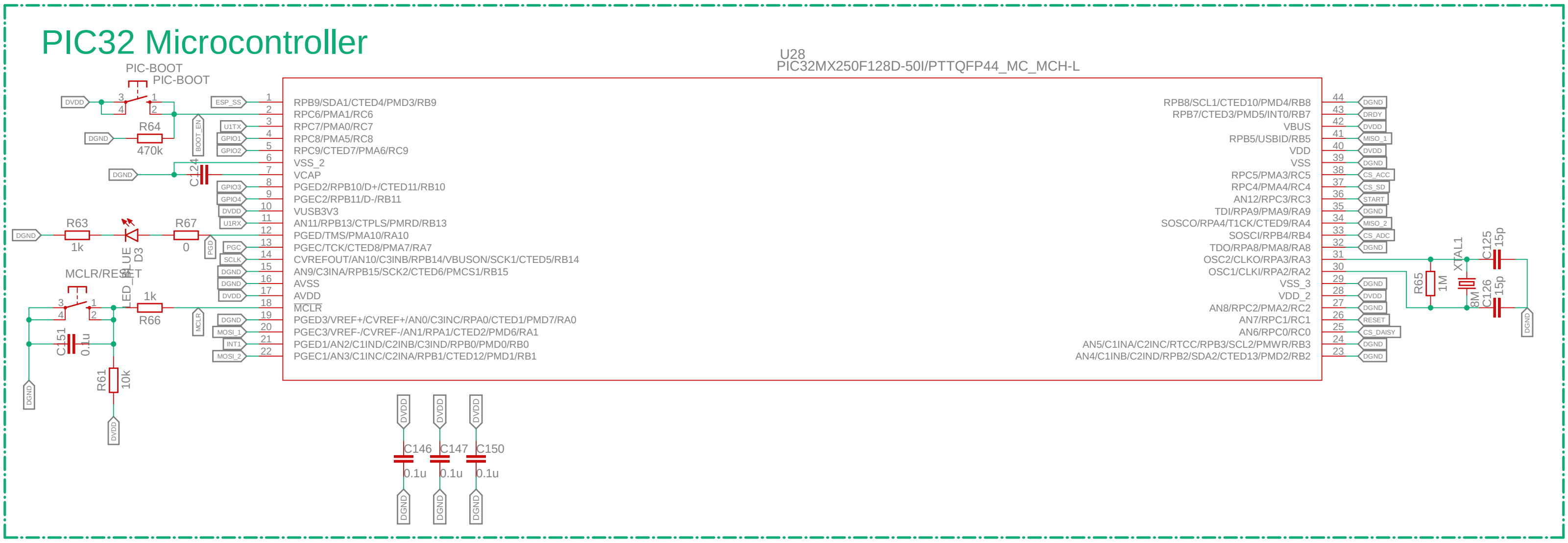
Microprocessor Interfacing

In the current iteration of the design, we have used 6 ADCs, an SD card module, an accelerometer, a WiFi module, and design inclusions for an external Daisy board interfacing. We plan on using four SPI buses for the above mentioned submodules. The breakup of SPI bus resource allocation is as follows-

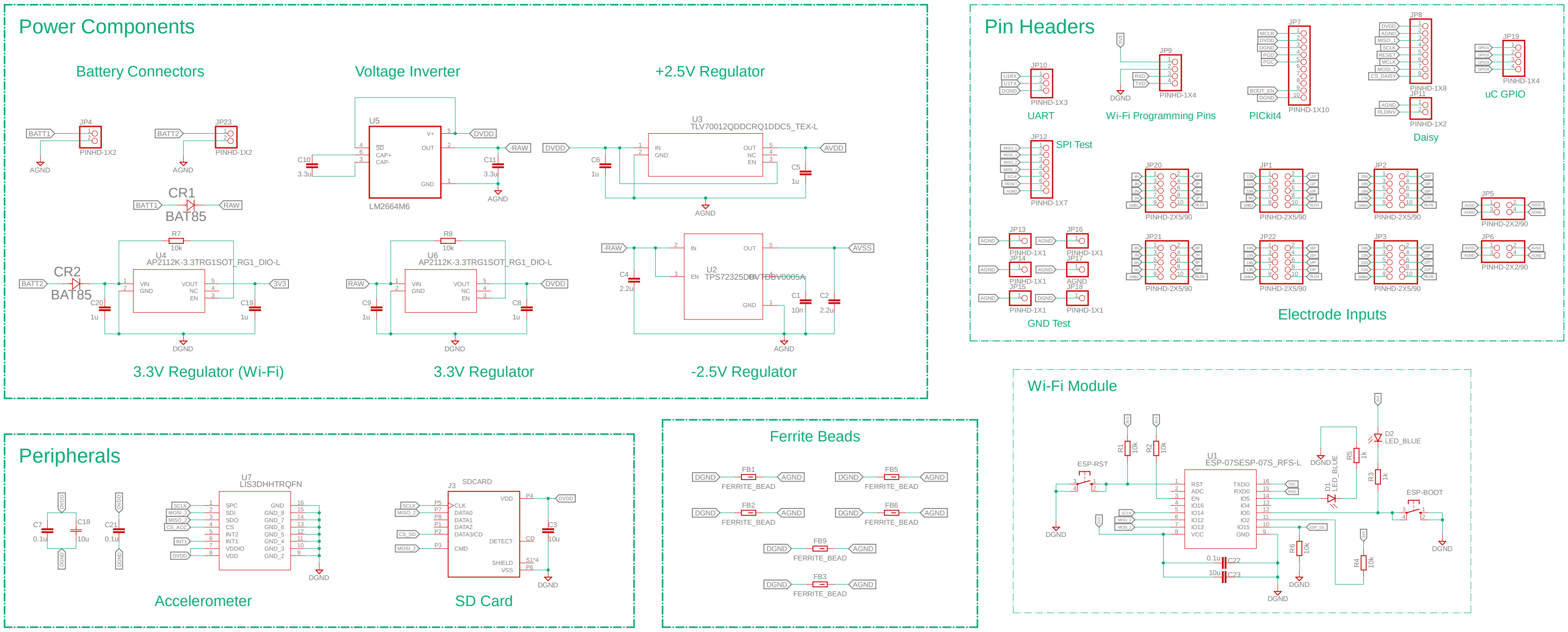
- **First SPI Bus** - For the six daisy-chained ADCs and External Daisy board interfacing
- **Second SPI Bus** - For the Wi-Fi module and peripherals such as SD Card slot and Accelerometer
- **PGEC and PGED** - For debugging and programming the controller

Circuit Schematic

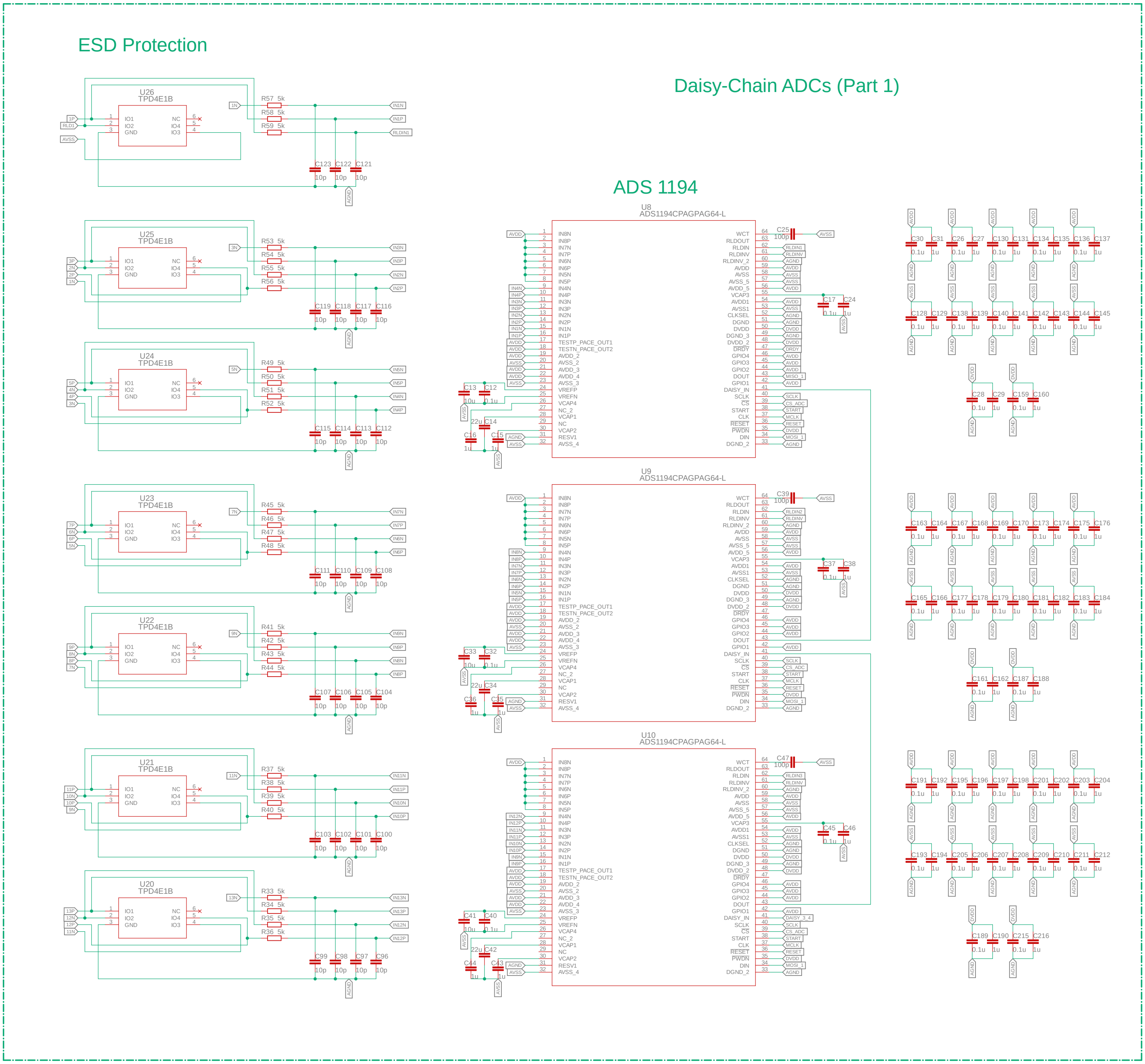
PIC32 Microcontroller

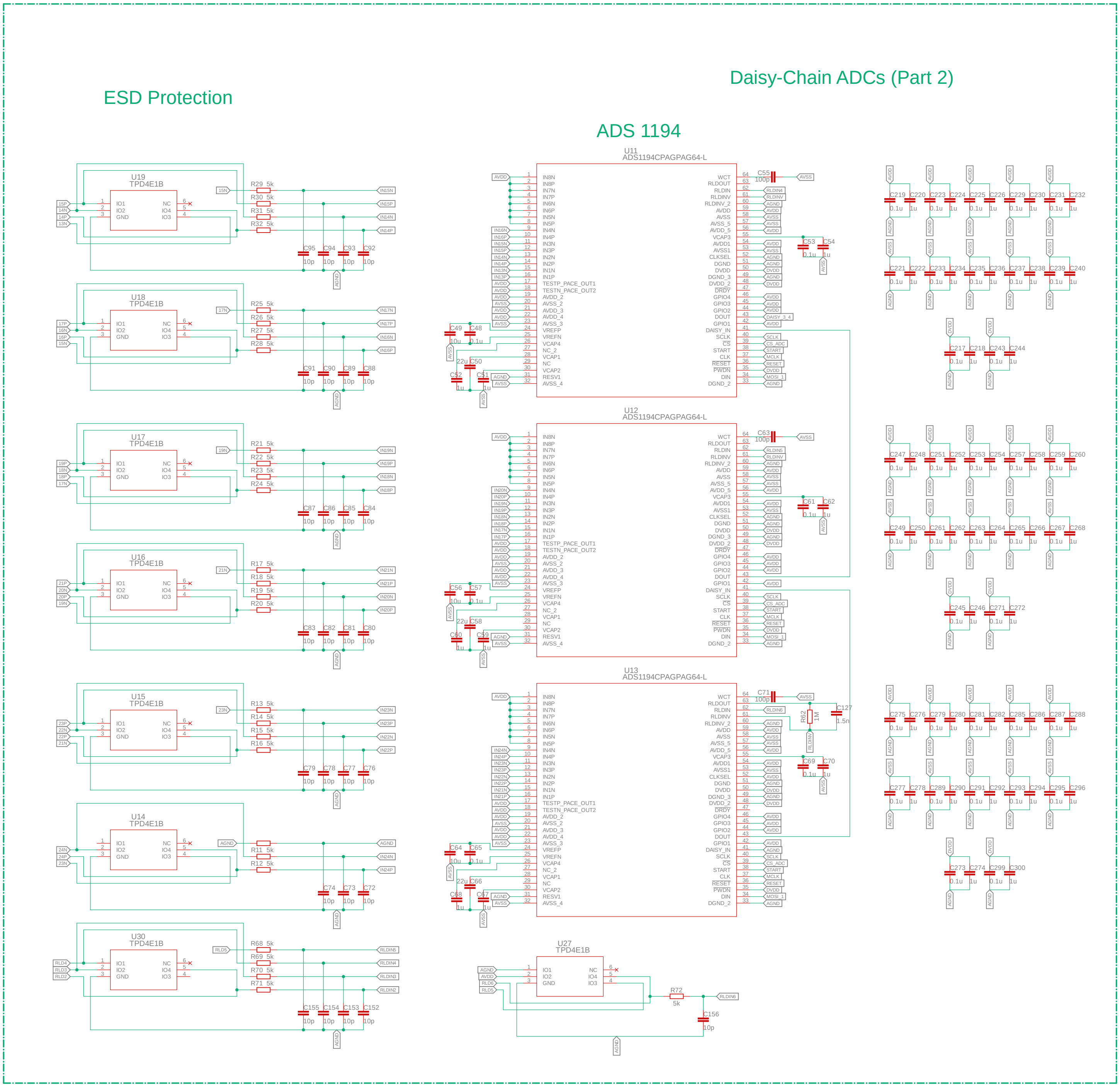


20-03-2023 17:56 f=0.88 C:\Users\AAYUSH~1\AppData\Local\Temp\Neutron\ElectronFileOutput\19640\sch-d1084b2b-3079-44c8-b09b-5ded0a6a41fa\Untitled.sch (Sheet: 1/1)



20-03-2023 17:57 f=0.52 C:\Users\AAYUSH~1\AppData\Local\Temp\Neutron\ElectronFileOutput\19640\sch-d1084b2b-3079-44c8-b09b-5ded0a6a41fa\Untitled.sch (Sheet: 1/1)


ESD Protection
Daisy-Chain ADCs (Part 1)
ADS 1194
U26 TPD4E1B
U25 TPD4E1B
U24 TPD4E1B
U23 TPD4E1B
U22 TPD4E1B
U21 TPD4E1B
U20 TPD4E1B
IO1 IO2 GND NC IO4 IO3
U8 ADS1194CPAGPAG64-L
U9 ADS1194CPAGPAG64-L
U10 ADS1194CPAGPAG64-L
IN8N IN8P IN7N IN7P IN6N IN6P IN5N IN5P IN4N IN4P IN3N IN3P IN2N IN2P IN1N IN1P
TESTP_PACE_OUT1 TESTN_PACE_OUT2 AVDD_2 AVSS_2 AVDD_3 AVDD_4 AVSS_3 VREFP VREFN VCAP4 NC_2 VCAP1 NC VCAP2 RESV1 AVSS_4
WCT RLDOUT RLDIN RLDINV RLDINV_2 AVDD AVSS AVSS_5 AVDD_5 VCAP3 AVDD1 AVSS1 CLKSEL DGND DVDD DGND_3 DVDD_2 DRDY GPIO4 GPIO3 GPIO2 DOUT GPIO1 DAISY_IN SCLK CS START CLK RESET PWDN DIN DGND_2
AVDD AVSS AGND DVDD RLDIN1 RLDINV SCLK CS_ADC START MCLK RESET MISO_1 MOSI_1 DAISY_3_4
R33–R59 5k
C96–C123 10p
C25 C39 C47 100p
0.1u 1u 10u 22u


ESD Protection
Daisy-Chain ADCs (Part 2)
ADS 1194
U19 TPD4E1B
U18 TPD4E1B
U17 TPD4E1B
U16 TPD4E1B
U15 TPD4E1B
U14 TPD4E1B
U30 TPD4E1B
U27 TPD4E1B
IO1 IO2 GND NC IO4 IO3
U11 ADS1194CPAGPAG64-L
U12 ADS1194CPAGPAG64-L
U13 ADS1194CPAGPAG64-L
IN8N IN8P IN7N IN7P IN6N IN6P IN5N IN5P IN4N IN4P IN3N IN3P IN2N IN2P IN1N IN1P
TESTP_PACE_OUT1 TESTN_PACE_OUT2
AVDD_2 AVSS_2 AVDD_3 AVDD_4 AVSS_3 VREFP VREFN VCAP4 NC_2 VCAP1 NC VCAP2 RESV1 AVSS_4
WCT RLDOUT RLDIN RLDINV RLDINV_2 AVDD AVSS AVSS_5 AVDD_5 VCAP3 AVDD1 AVSS1 CLKSEL DGND DVDD DGND_3 DVDD_2 DRDY GPIO4 GPIO3 GPIO2 DOUT GPIO1 DAISY_IN SCLK CS START CLK RESET PWDN DIN DGND_2
R29 5k R30 5k R31 5k R32 5k
R25 5k R26 5k R27 5k R28 5k
R21 5k R22 5k R23 5k R24 5k
R17 5k R18 5k R19 5k R20 5k
R13 5k R14 5k R15 5k R16 5k
R11 5k R12 5k
R68 5k R69 5k R70 5k R71 5k
R72 5k
R62 1M
C95 C94 C93 C92 10p
C91 C90 C89 C88 10p
C87 C86 C85 C84 10p
C83 C82 C81 C80 10p
C79 C78 C77 C76 10p
C74 C73 C72 10p
C155 C154 C153 C152 10p
C156 10p
C127 1.5n
C55 100p C63 100p C71 100p
C49 10u C48 0.1u C50 22u C52 1u C51 1u
C56 10u C57 0.1u C58 22u C60 1u C59 1u
C64 10u C65 0.1u C66 22u C68 1u C67 1u
C53 0.1u C54 1u C61 0.1u C62 1u C69 0.1u C70 1u
C219 C220 C223 C224 C225 C226 C229 C230 C231 C232
C221 C222 C233 C234 C235 C236 C237 C238 C239 C240
C217 C218 C243 C244
C247 C248 C251 C252 C253 C254 C257 C258 C259 C260
C249 C250 C261 C262 C263 C264 C265 C266 C267 C268
C245 C246 C271 C272
C275 C276 C279 C280 C281 C282 C285 C286 C287 C288
C277 C278 C289 C290 C291 C292 C293 C294 C295 C296
C273 C274 C299 C300
0.1u 1u
AVDD AVSS AGND DVDD
15P 14N 14P 13N 15N IN15N IN15P IN14N IN14P
17P 16N 16P 15N 17N IN17N IN17P IN16N IN16P
19P 18N 18P 17N 19N IN19N IN19P IN18N IN18P
21P 20N 20P 19N 21N IN21N IN21P IN20N IN20P
23P 22N 22P 21N 23N IN23N IN23P IN22N IN22P
24N 24P 23N IN24N IN24P
RLD4 RLD3 RLD2 RLD5 RLD6 RLDIN5 RLDIN4 RLDIN3 RLDIN2 RLDIN6
RLDINV DAISY_3_4 CS_ADC MCLK MOSI_1

PCB Layout

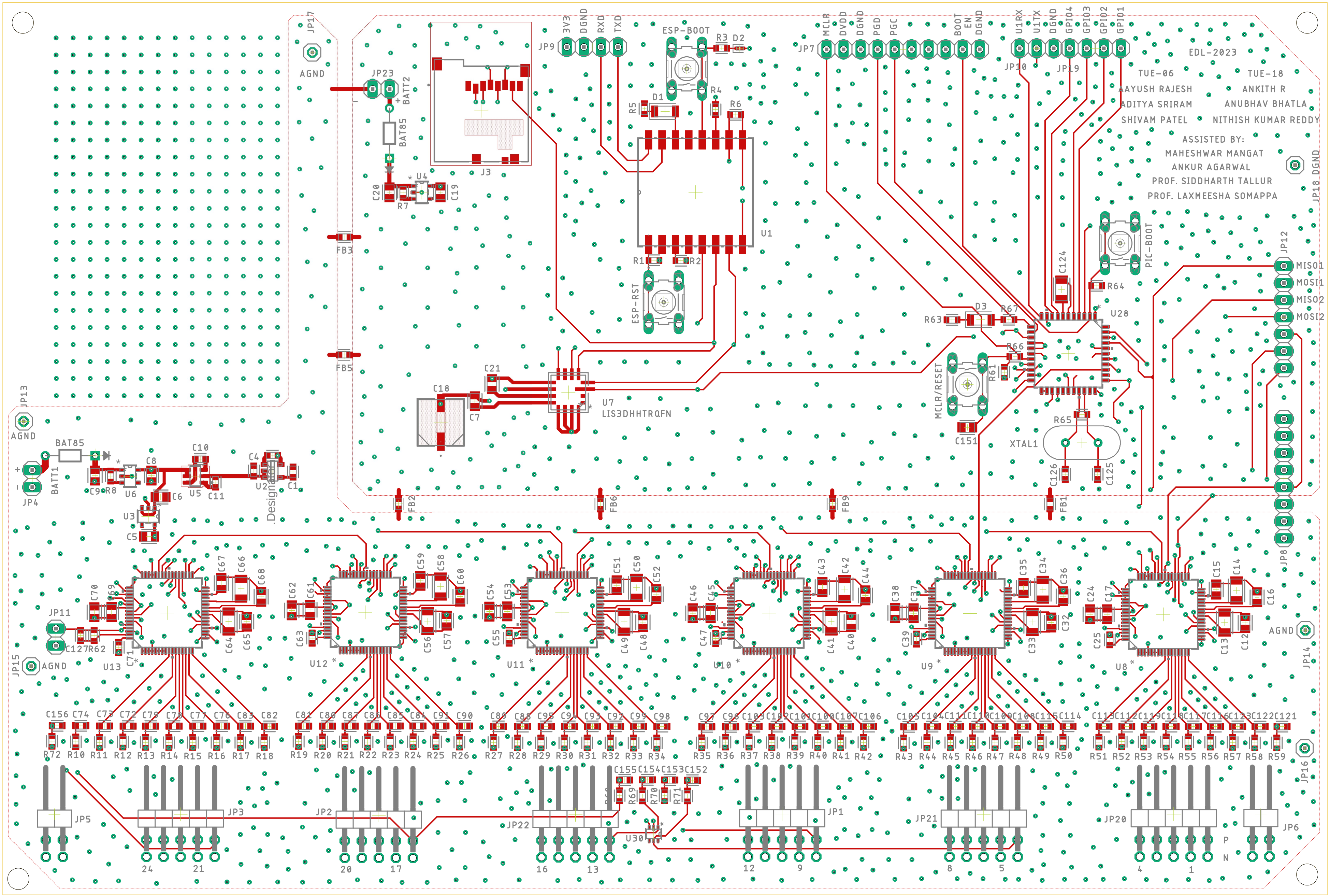
Due to the large-scale nature of our design, we have adopted a 4-layer PCB design. The top and bottom layers are where we house our components, and where the majority of the routing takes place. The inner layers are meant to function as power layers while allowing for routing as well, wherever it is not possible to do so using the outer layers. However, we have made sure to use these layers for routing as sparingly as possible, so as to ensure no deterioration in power signal quality.

Keeping in mind our requirements, as well as the manufacturer (PCBPower) specifics, we have vias in our design of type 1-2, 1-3, and 1-4. This allows connection between the top and bottom layers, as well as connections of the top and bottom layers with the inner layer, for both routing and power signal purposes. Stitching vias, which are through-hole vias, populate the board in regions wherever we have no components or routing, in order to connect our top and bottom layer copper fills. We have also provided a prototyping region in the top-left area of our board, where we do not have any components or copper fills.

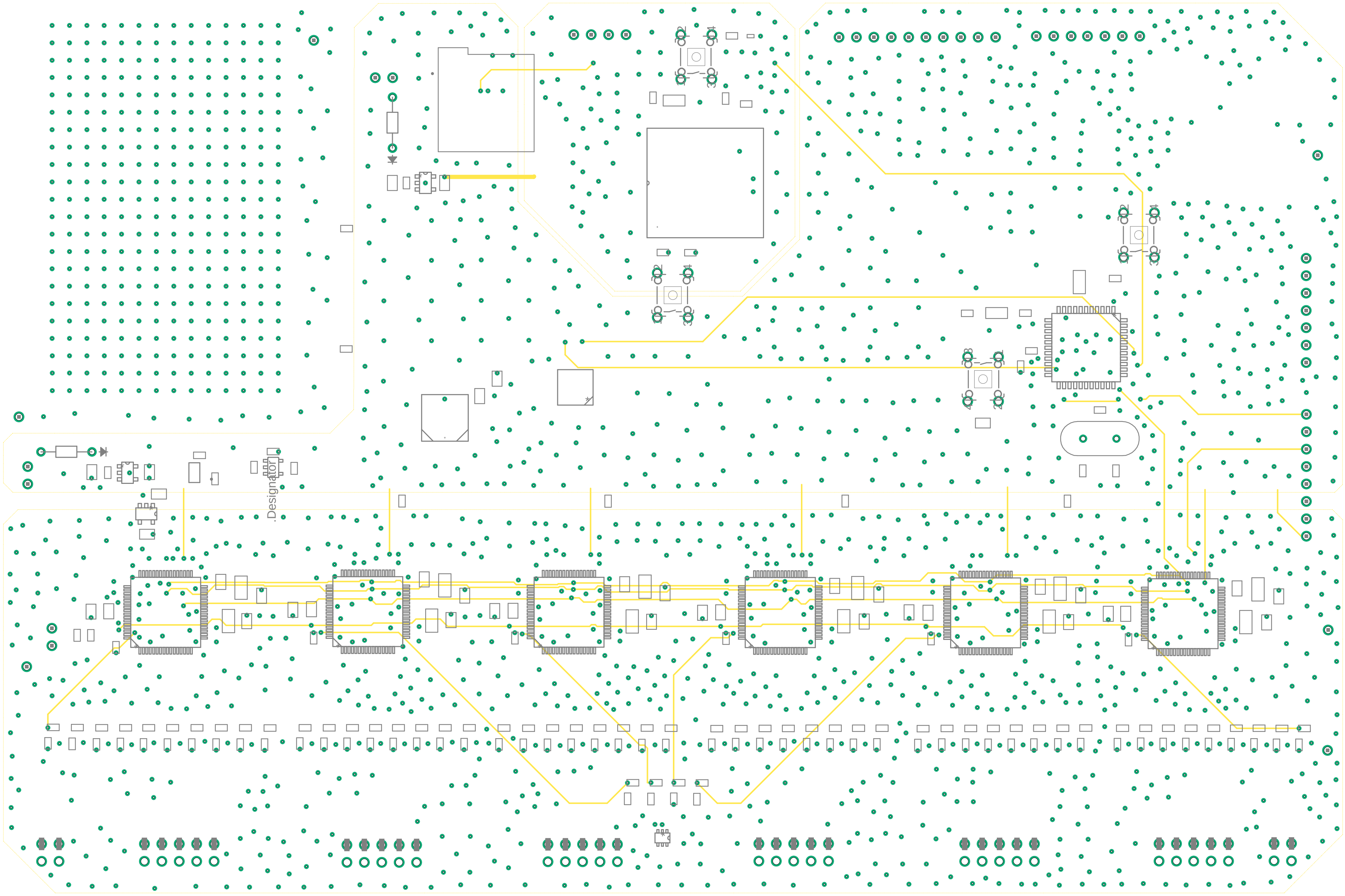
Coming to the copper fills, we have separate copper fills on the 4 layers corresponding to the Analog portion of the board (lower half) and the Digital portion of the board (upper half). Table 1 shows the exact nature of our layer stack. The demarcation of the exact regions can be seen on the following pages, which are arranged from the Top Layer all the way to the Bottom Layer.

Layer	Analog	Digital
Top	AGND	DGND
Layer 2	AVDD	DVDD
Layer 3	AVSS	DGND
Bottom	AGND	DGND

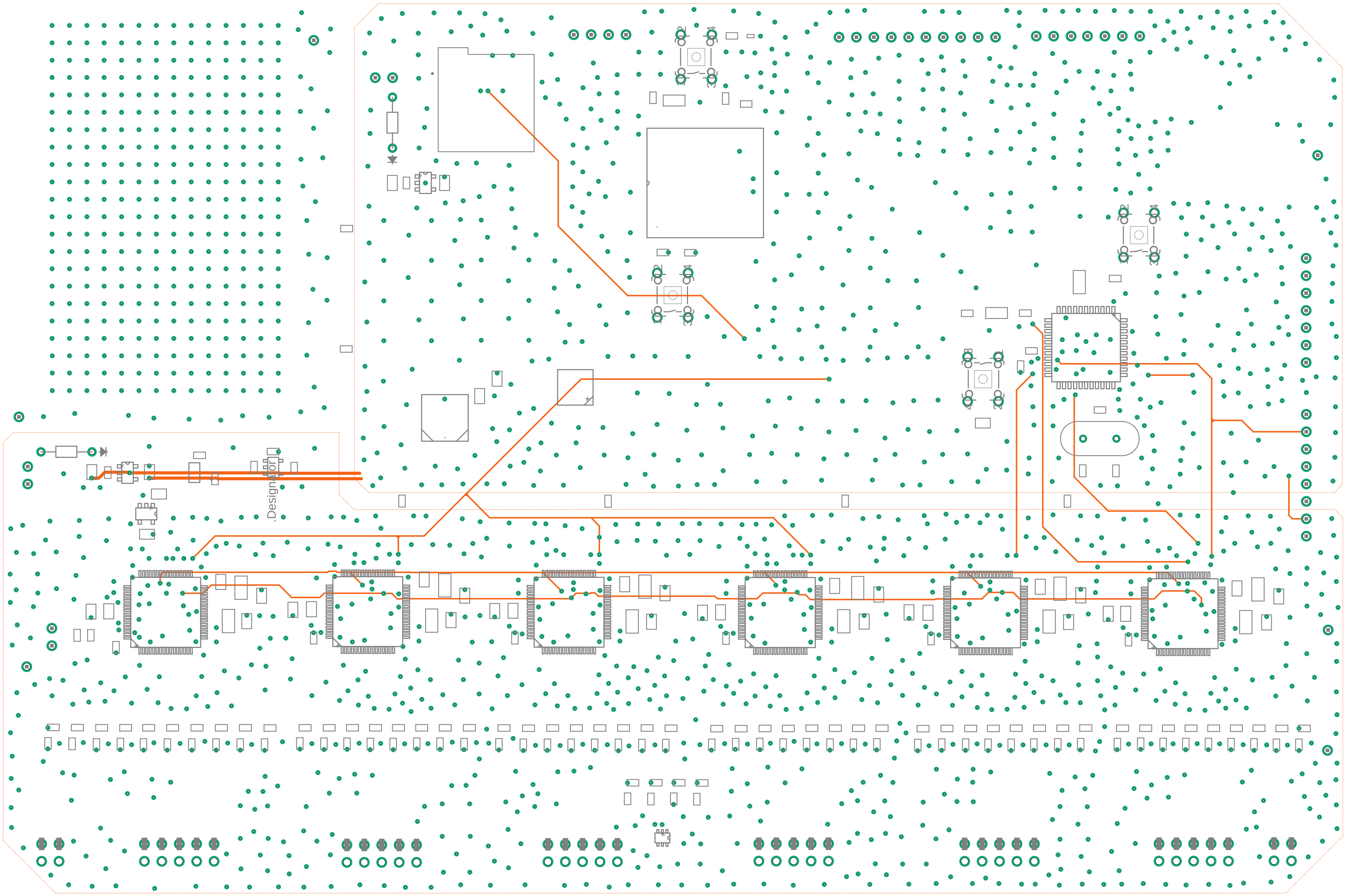
Table 1: Copper Fill Layer Stack



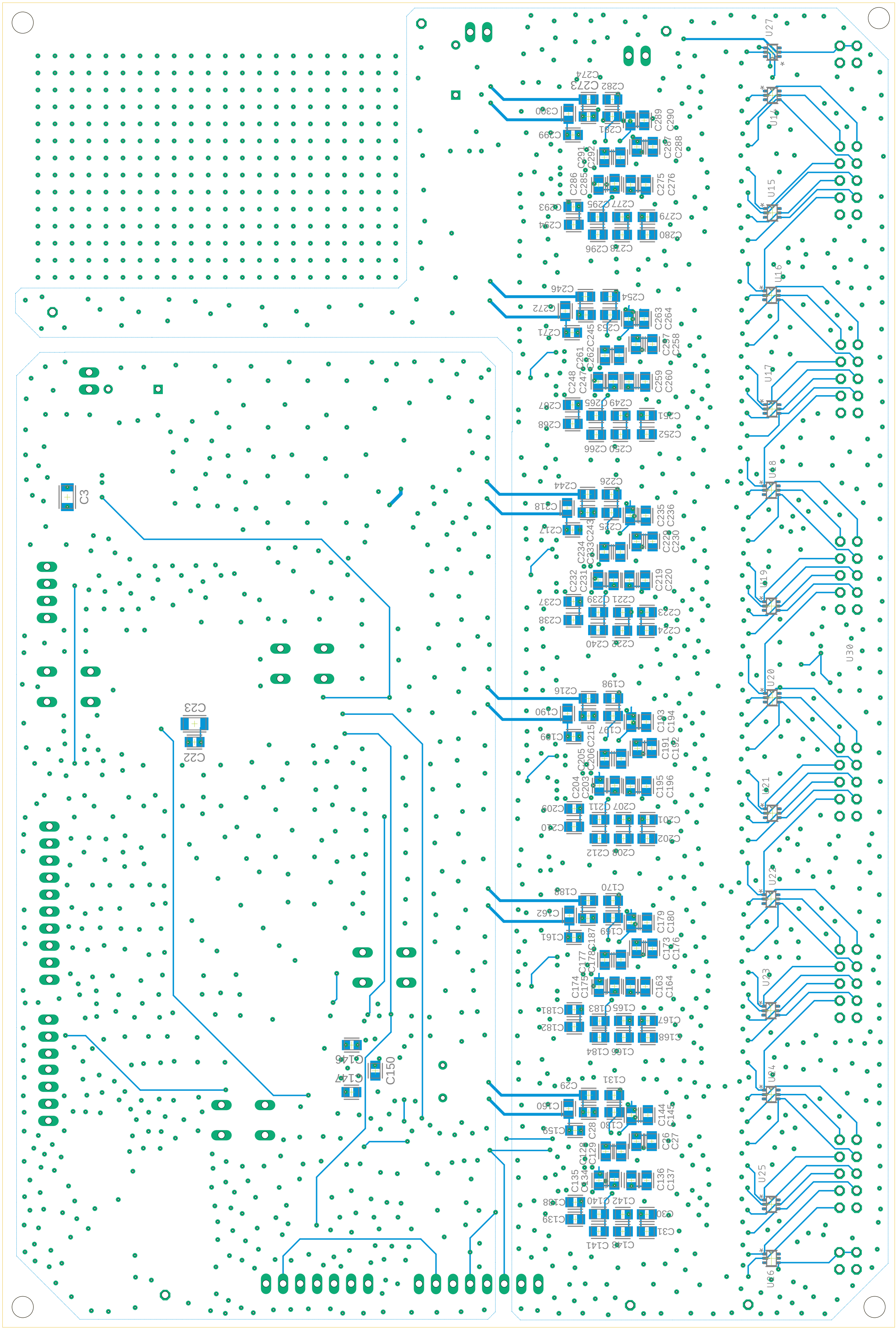
26-03-2023 09:45 f=1.38 C:\Users\AAYUSH~1\AppData\Local\Temp\Neutron\ElectronFileOutput\11924\brd-3ad7fbcb-1994-42f9-9d68-df653e6485a3\BCI_21_March_2023 v11.brd



26-03-2023 09:52 f=1.38 C:\Users\AAYUSH~1\AppData\Local\Temp\Neutron\ElectronFileOutput\11924\brd-3ad7fbcb-1994-42f9-9d68-df653e6485a3\BCI_21_March_2023 v11.brd



26-03-2023 09:53 f=1.38 C:\Users\AAYUSH~1\AppData\Local\Temp\Neutron\ElectronFileOutput\11924\brd-3ad7fbcb-1994-42f9-9d68-df653e6485a3\BCI_21_March_2023 v11.brd



26-03-2023 09:46 f=1.38 mirrored C:\Users\AAYUSH~1\AppData\Local\Temp\Neutron\ElectronFileOutput\11924\brd-3ad7fbcb-1994-42f9-9d68-df653e6485a3\BCI_21_March_2023 v

Top Layer

Almost all of our components are housed on the top layer. The lower half of the top layer is reserved for the Analog devices, such as the electrode connections, analog front-end filters, and the ADCs. Therefore, we have a copper fill of AGND in this region, which extends near the power regulators as well.

The top half of the layer has our digital devices, namely the microcontroller, WiFi chip, SD card, and accelerometer. Therefore, we have a DGND copper fill in this region. One notable connection is the accelerometer, which has pads under the device and is therefore applied with a stop mask surrounding it.

All our connectors - electrode connectors, battery connectors, test points, and headers for programming and debugging - are placed at the edge of the board. We had an empty region to the top-left of our board, and therefore made it a prototyping area by placing through-hole vias in the region. The Analog and Digital fills are connected by placing Ferrite beads along the region between their copper fills.

Layer 2

This layer is mainly a power layer and is also used minimally for routing (as can be seen from the layout). As mentioned in Table 1, the lower half has an AVDD copper fill, and the upper half has a DVDD fill. While the WiFi chip uses the same value as DVDD as the other digital components, it requires a higher power rating and therefore has a copper fill of its own.

Layer 3

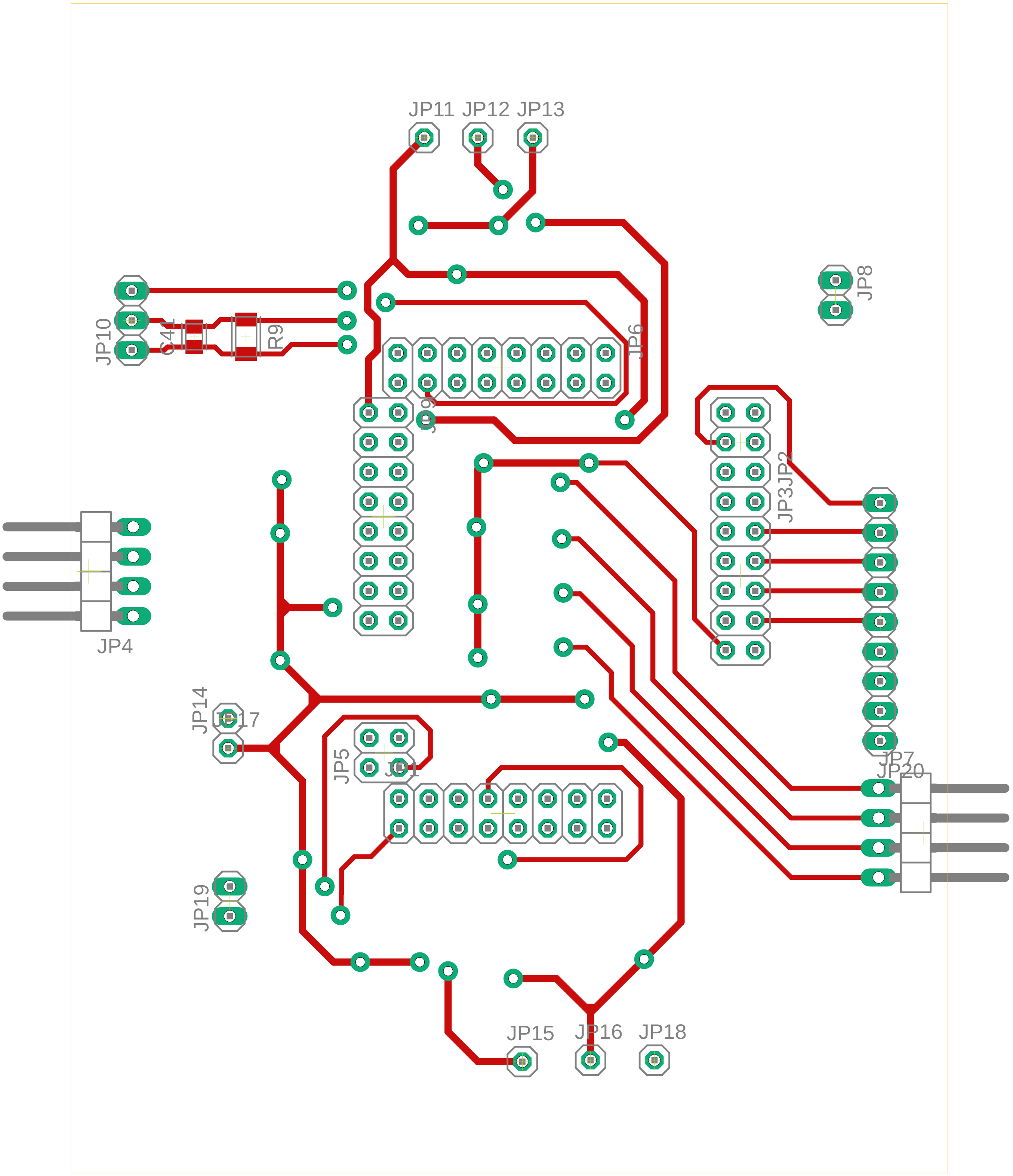
This layer is also predominantly a power layer and has fewer routes than Layer 2. The lower half has an AVSS fill, and since we did not require this layer for the power requirements of the digital half, we gave it a DGND fill.

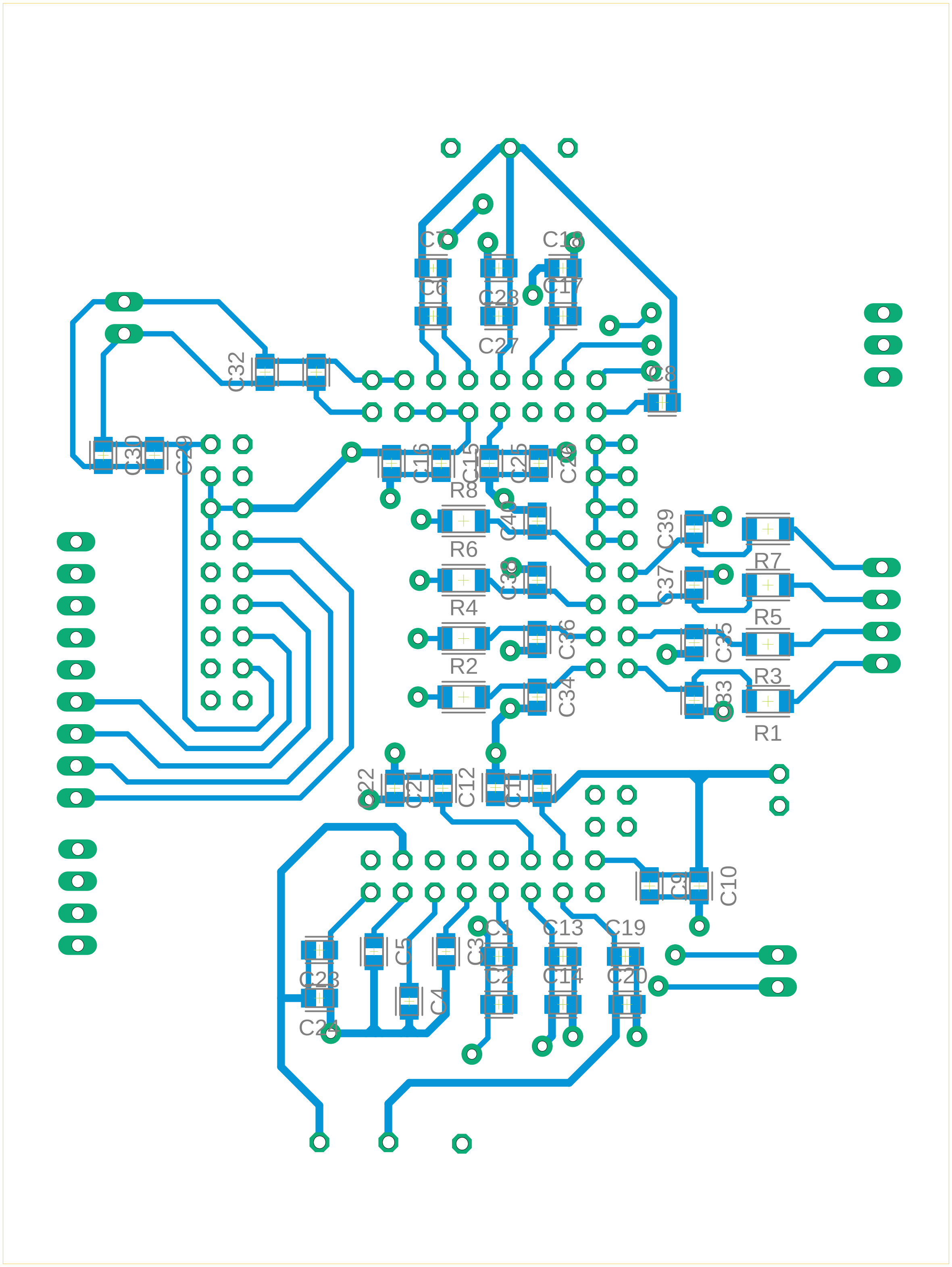
Bottom Layer

The bottom layer houses the ESD protection diodes and the majority of the decoupling capacitors of the devices. This is to ensure that the distance between the capacitor and the corresponding pin of the device is as short as possible, as the connection between them can be achieved using a very short trace from the pin followed by a through-hole via to the pad of the capacitor. As seen from the layout (which has been mirrored), the AGND and DGND copper fills of the bottom layer mirror those of the top layer for signal stability. The top and bottom layer fills are connected using stitching vias.

4-channel Module

Due to the large-scale nature of the end-to-end PCB, we also designed a modular PCB that houses a single ADS1194 placed on a breakout board. The modular PCB contains all the essential circuitry to operate the ADC, as well as interface it with the microcontroller and the EEG Acquisition circuitry. There are provisions to provide external AVDD, AVSS, DVDD, and AGND supplies to the board, as well as pin headers for SPI communication and electrode connections. Each module is compatible with 4-channel electrodes, and the nature of the design allows for it to be easily replicated in order to scale up the channel count through Daisy chaining of multiple ADC modules. The PCB layout for the modular ADC design is shown on the succeeding pages.


JP11
JP12
JP13
JP10
C4
R9
JP6
JP8
JP9
JP3
JP2
JP4
JP14
JP17
JP5
JP1
JP7
JP20
JP19
JP15
JP16
JP18

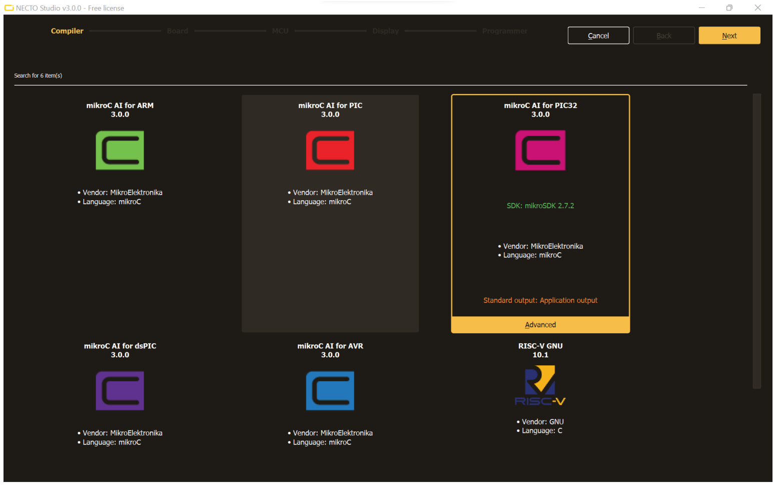


27-04-2023 14:27 f=2.20 mirrored C:\Users\AAYUSH~1\AppData\Local\Temp\Neutron\ElectronFileOutput\19296\brd-70aa9371

Software Setup

NECTOStudio Installation

NECTO Studio was used to program the PIC32MX Clicker board. The IDE can be installed from
(`https://www.mikroe.com/necto`). We have described the steps to create a new setup below:



mikroC AI for ARM
3.0.0
- Vendor: MikroElektronika
- Language: mikroC

mikroC AI for PIC
3.0.0
- Vendor: MikroElektronika
- Language: mikroC

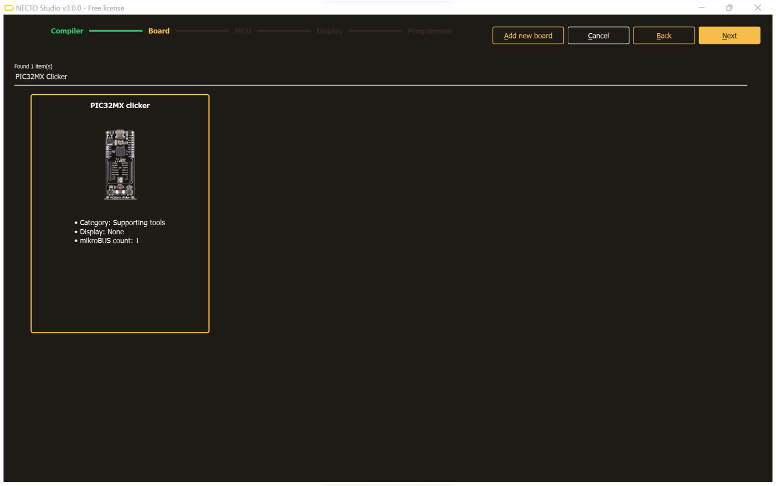
mikroC AI for PIC32
3.0.0
SDK: mikroSDK 2.7.2
- Vendor: MikroElektronika
- Language: mikroC

Standard output: Application output

mikroC AI for dsPIC
3.0.0
- Vendor: MikroElektronika
- Language: mikroC

mikroC AI for AVR
3.0.0
- Vendor: MikroElektronika
- Language: mikroC

RISC-V GNU
10.1
- Vendor: GNU
- Language: C

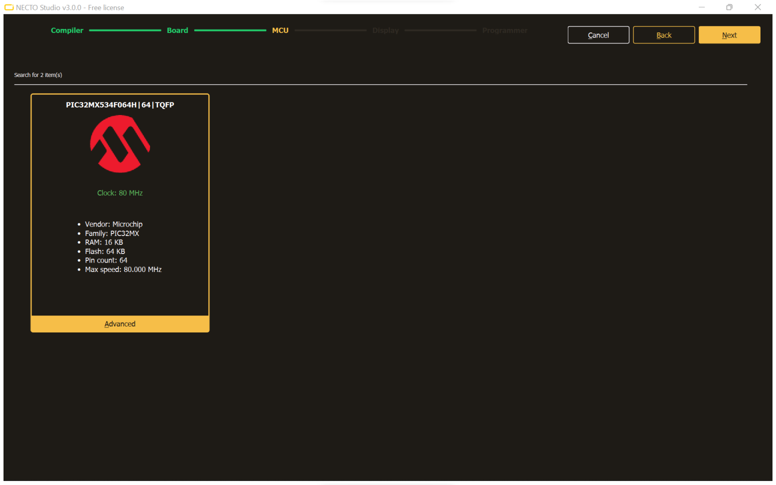


Found 1 item(s)

PIC32MX Clicker

PIC32MX clicker

- Category: Supporting tools
- Display: None
- mikroBUS count: 1

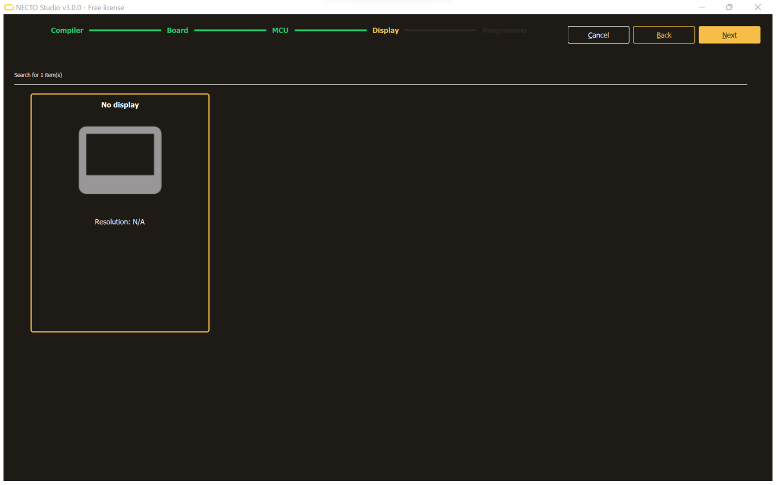


Search for 2 item(s)

PIC32MX534F064H | 64 | TQFP

Clock: 80 MHz

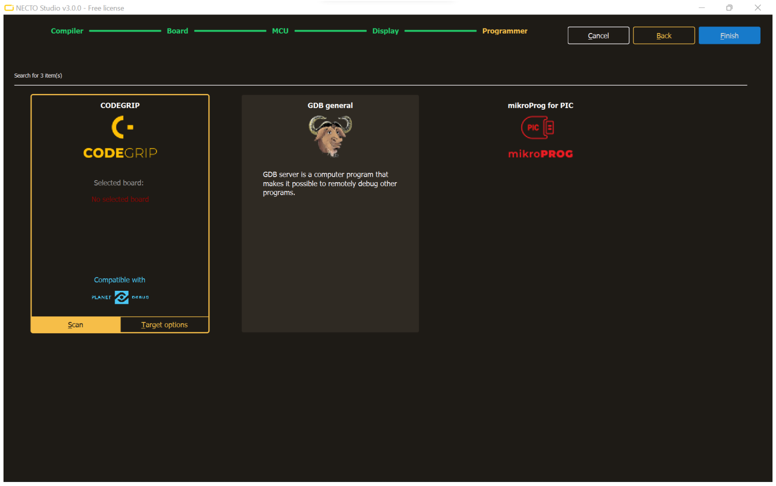
- Vendor: Microchip
- Family: PIC32MX
- RAM: 16 KB
- Flash: 64 KB
- Pin count: 64
- Max speed: 80.000 MHz



Search for 1 item(s)

No display

Resolution: N/A



Selected board: No selected board

GDB server is a computer program that makes it possible to remotely debug other programs.

Flashing Code on Microcontroller

1. Install mikroC PRO for PIC32 (`https://www.mikroe.com/mikroc-pic32#1200-lib-func-btn`).
2. Open the application -¿ go to tools and select the USB HID bootloader.
3. Connect the microcontroller to your laptop.
4. Click Connect. Browse for the HEX file and begin uploading.

USB-to-UART

Windows 10/11

1. Connect the USB-to-UART adapter to your PC.
2. Open the Device Manager and go to Ports.
3. If it shows '`PL2303HXA PHASED OUT SINCE 2012. PLEASE CONTACT YOUR SUPPLIER`', first download the device driver from the drive link: (`https://drive.google.com/file/d/14iQRifUeYDxDb8MPOXWJye5hQKZHVIbK/view`).
4. Unzip the folder and install the driver.
5. Click the message and go to driver details.
6. Select the update driver options and manually search for drivers from the list specified.
7. Choose the oldest option available and install it.
8. You should now be able to see the COM port correctly.

Linux

1. Open the terminal using `Ctrl+Alt+T`.
2. Install PuTTY: `sudo apt install putty`.
3. Connect the USB-to-UART adaptor to the laptop. Run `dmesg`, to find out which port the device is connected to. It should be something like `/dev/ttyUSB0`, `/dev/ttyUSB1`.
4. Change the permissions on the corresponding device file:
   `sudo chmod a+rw /dev/ttyUSB0`.

Acquiring Data

(Assuming you have configured USB-to-UART, the steps below are OS-agnostic)

1. Connect the USB-to-UART adapter.
2. Open a serial terminal software like PuTTY, Realterm.
3. Set the baud rate to 115200 and open the terminal.

Setting up Live Plotting Software

Python installation: We recommend using a virtual environment manager like Anaconda/Miniconda (https://docs.anaconda.com/free/anaconda/install/index.html) or `venv`. Follow the steps on the website to create a virtual environment. Install `pyqtgraph pyqt5 matplotlib pyserial` using `pip`. Connect the USB-to-UART adapter and run `python liveplot.py`.

Results

We conducted the demonstration by interfacing our ADC modular PCB with a PIC32MX Clicker board which transmitted data to the local device through UART. Electrodes were placed in specific locations of the 3D-printed headgear so as to perform different kinds of EEG acquisition experiments. Since the electrodes were single-ended, a body potential reference was provided through an ear clip which was used as the negative input for each of the four channels.

Blinking Activity Recording

The human forehead acts as an acquisition point for all neuro-electrical activity associated with facial muscles. Eye blinking is the most prominent facial action when recordability and magnitude of involved signals are concerned. We try to record the neuronal activity associated with the forehead plate electrodes (placed on channels 1 and 3). Our experimental setup contains hard and soft blinking cycles, spaced by few-second intervals. The hard blinks are clearly demarcated by a large magnitude of signals for a longer duration, whereas light blinking has lower magnitude signals with a small duration of activity. Also, note that time duration without blinking is characterized by zero value of recorded signals.

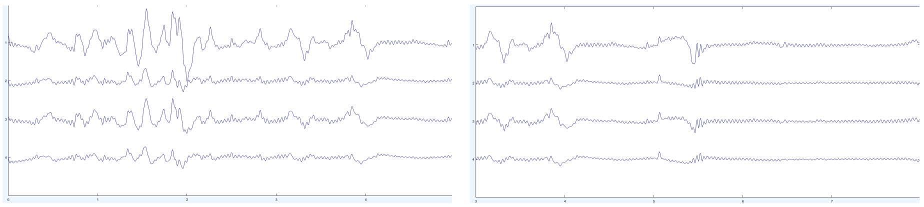


Figure 7: EEG Recording for Eye Blinking

Alpha Wave Acquisition

Alpha waves are primarily 8-12 Hz signals that are generated by rhythmic excitation of neurons in the occipital region of the brain when a person is meditating. Our experimental setup consists of the subject blinking a few times at the start of the acquisition and toward the end as well. There is a 10-second meditation period between the subject blinking. This is done to juxtapose the normal blinking signals with the alpha (meditation) waves that are being recorded. Please note that the electrodes placed towards the back of the subject's head show prominent and continuous acquisition of alpha waves.

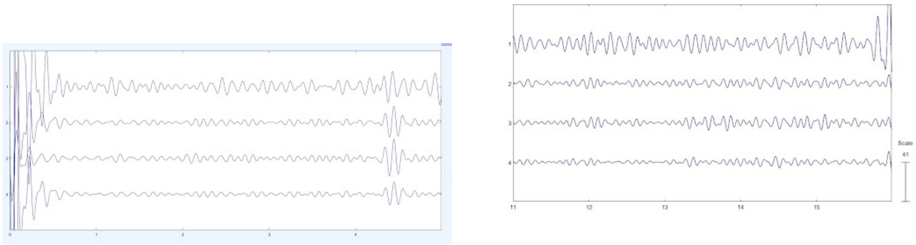


Figure 8: EEG Recording for Alpha Waves

Stimuli Response (via Strobe Lights)

The blinking activity recording and the alpha wave acquisition are activity-based electrical excitations of the neurons. We can also use the EEG apparatus to ascertain the subject's response to external stimuli. This is done by showing the subject 10Hz strobe lights and measuring the electrical activity in the occipital lobe near the back of the head. We observe that the excitations in the occipital region (associated with vision) match the frequency of the strobe light that the subject is exposed to.

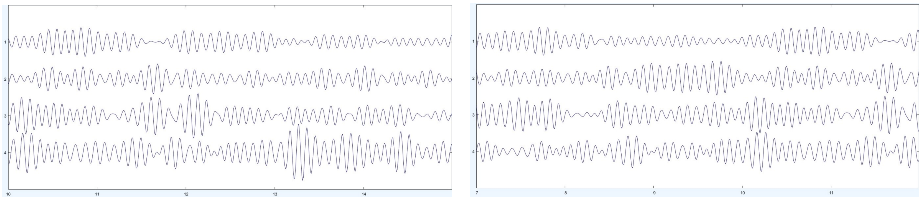


Figure 9: EEG Recording for 10Hz Strobe Light

Future Work

There are a few potential directions that the project can be taken further. Firstly, establishing a live plotting feature, instead of relying on the EEGLab software to post-process and plot recorded data, would be a major improvement. This would allow us to observe the EEG evolution in real time while the subject is responding to the stimulus, as well as compare the recording with the stimulus. Therefore, we do not need to rely on any experimental markers such as hard blinks which provide a sufficient, yet not entirely accurate, frame of reference. A challenge associated with this is to implement band-pass filters that are as selective as those implemented by MATLAB on EEGLab, as well as make these real-time. There exist functions in the `scipy` library on Python that may be up to this task.

The modular ADC PCBs designed for demo purposes are made to be Daisy chain compatible (we have provided the board files for the Daisy chain module as well). Therefore, this provides an efficient way to scale up the number of channels in use, perhaps in a stackable format, akin to FreeEEG32. While we were able to get the shift register action of the Daisy chain active, we were unable to resolve minor timing issues pertaining to the SCLK signals, which caused successive datastreams to interfere during the shifting action. Resolving these timing issues through software would allow the Daisy chaining of multiple ADC modules.

Finally, if the Daisy chain on the modular PCBs is figured out, that would allow implementation of the circuitry on the main PCB that we have designed. Since each ADC module implements the local circuitry around a single ADC on the main board, successfully chaining the ADC modules would act as a proof-of-concept for the main board, which has the capability to record 24 channels of data, with additional provisions such as Wi-Fi communication and SD card data storage.

Bill of Materials

Device Code	Generic Name	Quantity	Link
ADS1194	ADC	10	Mouser
PIC32MX250F128D-50I/PT	PIC Microprocessor	2	Mouser
ESP8266	Wifi Module	2	ElecComp
AP2112K-3.3TRG1	DVDD Voltage Regulator	4	Digikey
LM2664	Switched Cap V Regulator	2	Digikey
TPS72325	Neg. O/P Linear Regulator	2	Mouser
TLV70012QDDCRQ1	AVDD Voltage Regulator	2	Mouser
LIS3DHHTR	Accelerometer	2	Digikey
Sandisk 128GB MicroSD	SD Card	1	Amazon
TPD4E1B06DCKR	ESD Protection Devices	25	Digikey

Table 2: Bill of Materials for Main PCB

Device Code	Generic Name	Quantity	Link
ADS1194	ADC	10	Mouser
PIC32MX Clicker Board	PIC Microprocessor	1	Available in WEL
Comb Electrodes	Electrodes	8	Available in WEL
USB-to-UART Module	UART Connector	1	Available in WEL

Table 3: Bill of Materials for Demo PCB

References

1. OpenBCI Documentation
2. https://arxiv.org/ftp/arxiv/papers/1808/1808.03711.pdf
3. ADS1194 Datasheet
4. PIC32MX250F128B/D Datasheet
5. ESP8266EX Datasheet
6. AP2112 Datasheet
7. LM2664 Datasheet
8. TPS72325 Datasheet
9. TLV700 Datasheet
10. LIS3DHHTR Datasheet
11. TPD4E1B06DCKR Datasheet