MAUDLIN: INTERFACE FOR RANDOM WALKS AND BEYOND - EURORACK MODULE FOR UBIQUITOUS MUSIC

JOHN HARDING[1]

J.HARDING@ULSTER.AC.UK

ABSTRACT

This paper offers an expanded review of work presented at the ubimus 2025 conference, detailing the MAUDLIN Eurorack module inspired by the "drunk" object in Cycling 74's Max/MSP. MAUDLIN introduces a novel interface for generating controlled randomness in ubiquitous music (ubimus) practices by leveraging an unconnected Analog to Digital Converter (ADC) pin as a natural entropy source to augment pseudo-random sequences constrained within user-defined boundaries. Designed for modular synthesisers, the module integrates control voltage (CV) inputs, potentiometers, and an display for real-time interaction and feedback, aligning with ubimus principles of accessibility and creative flexibility. We detail the module's de sign principles and the use of environmental (EMF) noise to seed and perturb randomisation, we also discuss the potential to enhance musical creativity through constrained random walk outputs. As an open-source hardware and software development, MAUDLIN aims to democratise access,

1. Ulster University - Derry/Londonderry

fostering collaborative innovation in the ubimus and Eurorack communities while serving as a potential platform for expanded applications in embedded technology contexts.

1 INTRODUCTION

Ubiquitous music (ubimus) emphasises accessible, context-aware, and collaborative musicmaking, leveraging everyday technologies to foster creativity [Keller et al., 2014]. Randomness has long been a creative force in music, from the aleatoric compositions of John Cage to modern generative systems that introduce constrained unpredictabil ity [Barton and Fritz, 2024]. Constrained randomness, where random processes operate within defined boundaries, balances serendipity with intentionality, enabling novel out comes to be accessible to diverse practitioners. In ubimus, such approaches lower barriers to musical expression, aligning with its ecologically grounded ethos of creativity emerging from human-technology-environment interactions.

The MAUDLIN Eurorack module aligns with these principles by providing a compact and interactive tool for generating controlled randomness in modular synthesis. Inspired by the 'drunk' object in Max / MSP [Cycling '74, 2025], which implements a Markov chain random walk [Konstantopoulos, 2009], MAUDLIN extends this concept to the hardware domain, offering a tactile interface for live performance and composition. A key innovation is centred around the use of an unconnected ADC pin as a source of environmental (EMF) noise, capturing inherent electrical fluctuations to augment the production of constrained random sequences. This approach offers a supplementary approach to digital pseudo-random number generators (PRNG) [Knuth, 1998] or the more traditional analog transistor-based white noise circuits [Moog Music Inc., 1978], providing a system that enhances musical expressivity [Barton and Fritz, 2024].

This paper presents MAUDLIN's design, implementation, and relevance to ubiq uitous music practice, emphasising its role in democratising creative tools through open source development and its capacity to foster intuitive, unpredictable but constrained musical interactions. By bridging analog and digital domains, MAUDLIN aims to serve as a model for future embedded system creativity, with potential applications beyond music in generative art and networked systems [Turchet et al., 2018].

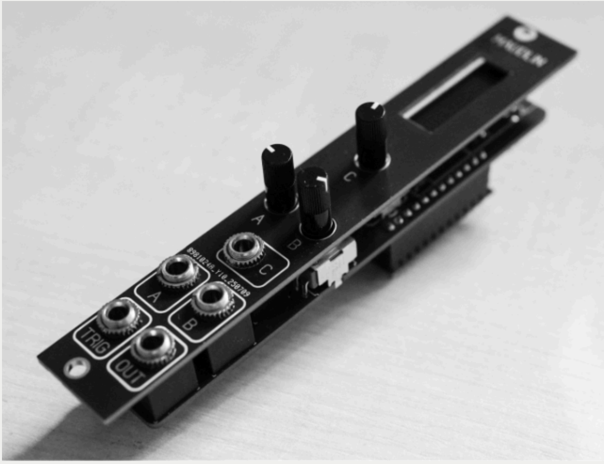

FIGURE 1: MAUDLIN HARDWARE V 1.0

2 DESIGN AND IMPLEMENTATION

The MAUDLIN unit in standard Eurorack format, depicted in Figures 1 and 2 and authored by John Harding in 2025, is specified as follows:

2.1 ARCHITECTURE

The core functionality of this development is inspired by the random walk functionality of the ’drunk’ object [Cycling ’74, 2025], where the values are produced in response to a trigger and drift within a user-defined range, and where the current value is probabilistically related to those that preceded it, reflecting the dynamic processes of analog computers [Lazzarini and Timoney, 2020]. An Arduino Pro Mini microcontroller [Arduino, 2025] is selected for its low cost, small footprint, and ease of use. An external clock/trigger input allows synchronisation, and three control voltage (CV) inputs labeled; A, B and C allow for modulation via external CV sources while associated potentiometers; A, B and C allow for manual adjustment. Finally, a Digital to Analog (DAC) converter produces the CV output [Microchip Technology Inc., 2009].

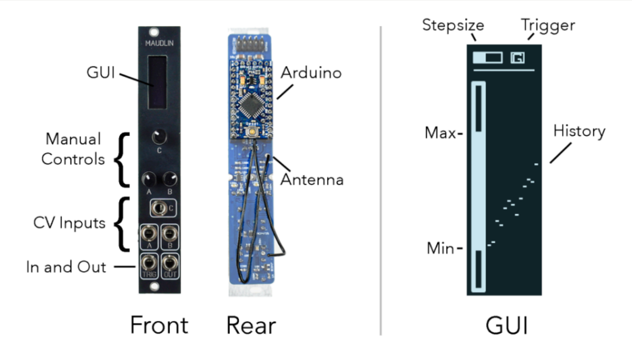

FIGURE 2: MAUDLIN HARDWARE V 1.0 AND GUI

The unconnected or floating ADC pin with an optional attached antenna, shown in Figure: 2, serves as an entropy source, capturing electronic/analog environmental (EMF) noise to seed the random walk in terms of directionality and step size, which is con strained within user-defined boundaries and uses reflection to maintain continuity similar to the nonlinear dynamics of analog synthesis systems. [Lazzarini and Timoney, 2020]. The interface prioritises accessibility, allowing users to intuitively implement controlled randomness without extensive technical knowledge, a core ubimus tenet.

While initial focus is placed on the Markov chain random walk application under current discussion, the hardware is intentionally designed to facilitate a series of other functions in the future, ensuring that MAUDLIN is adaptable for various musical uses and settings. The features allow the hardware to be used for various optional modes in future, adding to the versatility across diverse musical contexts.

2.2 HARDWARE SPECIFICATION

To conform to the Eurorack ethos of minimal horizontal width (HP), MAUDLIN is de signed with surface-mount electronic components, occupying 8 HP. It adheres to Eurorack power constraints (±150mA per ±12V rail)[1]. Components include the Arduino Pro Mini, MCP4725 DAC, SSD1306 OLED, MCP6002 rail-to-rail op-amps for CV input and output conditioning, and 3.5mm jacks for CV and gate inputs and one CV output. The front panel features a cutout for OLED visibility with labeled controls. The interface prioritises accessibility, allowing users to shape randomness intuitively. [Keller et al., 2014]. Full technical details are available through the online repository: [Harding, 2025a].

3 FIRMWARE

The firmware processes inputs to generate a random walk seeded by EMF noise, scaled and directed within user-defined boundaries, and interfaced with an MCP4725 Digital to- Analog Converter (DAC) [Microchip Technology Inc., 2009] for CV output. The initial design focuses on a range of 0V to 5V DC output CV, with future development planned for ±5V output via an open-ended conditioning circuit. The SSD1306 OLED [Solomon Systech Limited, 2012] provides real-time feedback of control pa rameters and output values. The system integrates CV inputs (step size, range, off set) and corresponding potentiometers, and the module adheres to Eurorack standards [Doepfer Musikelektronik, 2023].

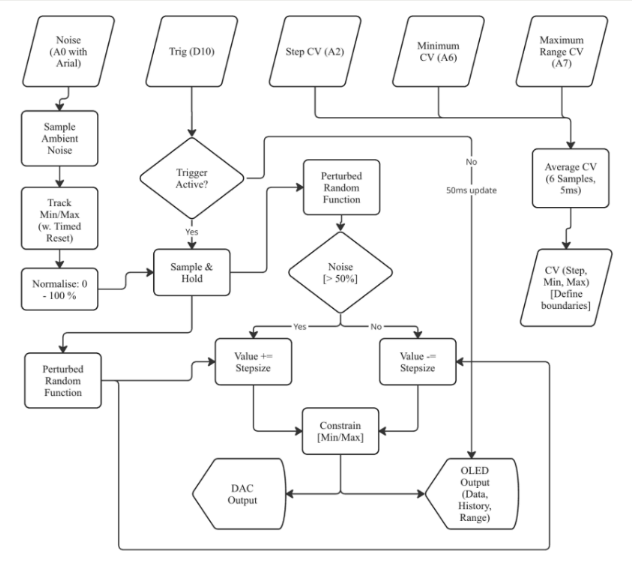

FIGURE 3: MAUDLIN: DRUNK INSPIRED, MARKOV-CHAIN. SIMPLIFIED LOGICAL FLOW DIAGRAM

3.1 FIRMWARE: LOGICAL FLOW AND STRUCTURE

As depicted schematically in Figure: 3, environmental (EMF) noise is captured via an unconnected or floating ADC pin (A0), with an optional 5-10 cm antenna attached to the rear

1. Doepfer Eurorack Standard [Doepfer Musikelektronik, 2023]

of the PCB, serving as an entropy source. A double read protocol (discarding the first A0 read after grounding A1, with a 10μs delay) minimises crosstalk between adjacent channels. On a trigger input (via D10), the noise is sampled and held, with normalisation to a 0-100 percent range using dynamically tracked min/max boundaries, reset every 1000 samples (~5 s) following a 20-sample (~100 ms) stabilisation period. For low noise variation (range <50), the normalised value is amplified (multiplier = 100 / range). The normalised noise percentage offsets a PRNG base-direction value, constrained to 0-100%, to determine the step direction. Step magnitude is a similarly perturbed random value from 0 to stepsize and if EMF readings are below 20, the system offers a fallback mechanism to pure PRNG to ensure reliable performance across lower measured EMF environments. This augmentation process blends PRNG reliability with noise entropy for robust randomness across varied EMF environments, with the walk value constrained to [minimum, maximum] before DAC output. A 128x32 SSD1306 OLED displays a step size bar, min/max slider, output history, and gate indicator, updated every 50 ms. The trigger-to-output response time ranges from 2.7-3.4 ms, supporting rates up to 294 Hz, with nonblocking C++ routines ensuring low-latency operation. Alternate firmware modes may support increased rates depending on the coding complexity.

3.2 FIRMWARE BIAS AND AUTO-ADJUST

The latest Drunk V.5 firmware iteration[2] enhances the Markov chain random walk with bias correction and auto-adjustment to optimise noise distribution for a less biased

[FormattingFormatting]

ran domisation of step size and directionality. The module retains its core structure: CV inputs (STEP_CV, OFFSET_CV, RANGE_CV) control step size (0-512), minimum (0-4095), and maximum (0-4095), respectively, with noise from NOISE_PIN pertubing direction and step size via a normalised 0-100% value. A gate trigger (TRIG_PIN) updates the DAC output, constrained within user-defined boundaries, and the OLED displays a step bar, slider, history, and gate indicator (Figure: 2). Key features include:

- **Bias Adjustment:** An autoBias variable (initially 30) is applied to the normalised noise percentage post-mapping, adjusting the 0-100% range to correct for skew (e.g., shifting more values above 50% for positive steps). This is constrained between 0 and 50 to prevent overcompensation.

- **Auto-Adjust Mechanism:** The system calculates a running mean of raw (pre-bias) percentNoise over 100 held samples. If the mean deviates beyond a 5% hysteresis band (below 45% or above 55% from the target 50%), autoBias increments or decrements by 1, ensuring the distribution centers dynamically.

- **Robustness:** Full initialisation of averaging buffers to 512, a watchdog timer resetting after 8 seconds, and improved min/max resets (falling back to 100/900 if invalid) enhance reliability. [FormattingFormatting]

2.https://bitbucket.org/Nonzerojohn/maudlin/src/main/Maudlin%20Code/

- Performance: Non-blocking timing (5ms sampling, 50ms display) and ADC crosstalk mitigation (GND_PIN resets, double-read) maintain low latency, suitable for real time Eurorack use.

This solution, detailed further in the following sections, preserves the original intent while adapting to environmental noise variations.

3.3 NOISE AUGMENTATION FOR VARIABLE ENVIRONMENTS

The development of the MAUDLIN Drunk inspired firmware highlighted challenges in utilising environmental electromagnetic (EMF) noise as an entropy source to drive the Markov chain random walk. A series of tests were conducted which revealed a bias towards negative steps due to the limited dynamic range (~0-250 ADC units) and frequent zero values, resulting from its lower end skew during normalisation to a range of 0- 100% via dynamically tracked min/max values. This constrained variability, tested to vary across environments, prompted a hybrid solution. To mitigate inherent biases—such as skew toward lower values observed in initial testing—the system employs dynamic min/max tracking and periodic resets, ensuring the noise distribution adapts to varying conditions. Hybrid techniques further enhance reliability: pseudo-random number generation (PRNG) [Knuth, 1998] is augmented with true noise, which seeds both step-size and directionality, blending determinism with organic fluctuation. The aforementioned auto-adjustment mechanism monitors the mean of normalised noise values, incrementally tuning bias (initially set at 30) to center the distribution around the midpoint, promoting balanced entropy without over-reliance on either source. This approach addresses practical challenges like zero-dominance and, if noise measurements fall below 20, the system falls back to utilising pure PRNG for both step size and direction. This ensures consistent performance across varied EMF environments: noise adds subtle perturbations in low-variation environments and dominates in high, blending sources for robustness across contexts.

3.4 EXAMINATION OF NOISE CONDITIONING: THE UPDATENOISE(); FUNCTION

To ensure a reliable entropy source, the updateNoise() function processes raw noise data from the unconnected ADC pin (A0), adapting to environmental electromagnetic field (EMF) fluctuations. The relevant function is presented below:

```
void updateNoise() {
  noiseValue = rawNoise;

  // Update min/max tracking
  if (noiseSampleCount >= STABILIZATION_SAMPLES) {
    if (rawNoise < minNoise) minNoise = rawNoise;
 if (rawNoise > maxNoise) maxNoise = rawNoise;
 } else {
 noiseSampleCount++;
 }
 resetCounter++;
 if (resetCounter >= RESET_INTERVAL) {
 minNoise = (rawNoise > 0) ? rawNoise : 100;
 maxNoise = (rawNoise < ADC_RESOLUTION) ? rawNoise : 900;
 resetCounter = 0;
 }
}
```

This function initialises noiseValue with the latest raw ADC reading and implements a stabilisation period (STABILISATION_SAMPLES = 20) before updating the min/max boundaries. Once stabilised, it tracks the minimum and maximum noise values dynamically, ensuring the normalisation range reflects current conditions. The reset mechanism, triggered every 1000 samples (~5s), re-initialises these boundaries with fall backs (100/900) if the current noise is invalid, preventing stale distributions. This adaptive approach mitigates biases (e.g., zero-skew) observed in initial tests. Future optimisations could involve exponential moving averages for smoother tracking, though this increases computational overhead on the 16MHz Pro Mini.

3.5 EXAMINATION OF EMF PERTURBATION MECHANISM

The EMF perturbation mechanism in MAUDLIN's Drunk firmware represents an integration of environmental entropy into a controlled random walk. As shown graphically in Figure 3: the perturbation is applied to both the direction and step size, using the bias adjusted and normalised noise percentage (0-100%) derived from the ADC pin (A0). For direction, the base PRNG value (random(0, 101)) is offset by (percentNoise - 50), restricted to 0-100, allowing EMF noise to subtly influence the probability of posi tive or negative steps. For step size, the base random value (random(0, stepsize + 1)) is perturbed by a noise-derived offset, calculated as map(percentNoise, 0, 100, -stepsize/10, stepsize/10), and constrained to 0-stepsize. This scaling ensures the perturbation remains proportional to the step size, typically ±51 for a maximum stepsize of 512, adding organic variation without exceeding bounds.

Example: Direction Perturbation

The base PRNG direction (random(0, 101)) is offset by percentNoise - 50, constrained to 0-100, allowing EMF noise to bias the probability of positive or negative steps.

```
int baseDirection = random(0, 101); // PRNG base for direction int
augmentedDirection = baseDirection + (percentNoise - 50); // Noise
augments (offsets) PRNG
augmentedDirection = constrain(augmentedDirection, 0, 100);
```

Example: Step Size Perturbation

The step size is calculated using random(0, stepsize + 1) and per turbed by a noise-derived offset, scaled via map(percentNoise, 0, 100, -stepsize/10, stepsize/10). This keeps perturbations proportional (e.g., ±51 for stepsize=512), constrained to 0-stepsize.

```
int randomStep = random(0, stepsize + 1); // Random step 0 to stepsize
int noiseOffset = map(percentNoise, 0, 100, -stepsize/10, stepsize/10);
randomStep = constrain(randomStep + noiseOffset, 0, stepsize);
```

3.6 EXAMINATION OF FALLBACK AND AMPLIFICATION

[Indent]The implementation includes a fallback mechanism for low noise ranges (< 20), which mirrors the direction process. If currentRange < LOW_NOISE_THRESHOLD, both direction and step perturbations revert to pure PRNG values, ensuring functionality in environments with minimal EMF interference (e.g., 0-19 range). Amplification stretches noise variation for ranges 20-50, tested to handle transitions from low to high noise (0- 260+), validated through oscilloscope measurements showing consistent step distribution. This hybrid approach balances computational efficiency with environmental randomness, critical for adaptability.

[Indent]For EMF low noise ranges (<20), the mechanism falls back to pure PRNG for both direction and step size, ensuring reliability in low-EMF conditions (e.g., 0-15 range). For ranges 20-50, amplification stretches noise variation, validated by oscilloscope tests showing consistent step distribution across high-noise transitions (0-260+).

Example: Fallback Mechanism

```
 int currentRange = maxNoise - minNoise;
 if (currentRange < LOW_NOISE_THRESHOLD) {
 rawPercent = random(0, 101); // Pure PRNG for percent }else if
(currentRange < MIN_NOISE_RANGE && currentRange >=
LOW_NOISE_THRESHOLD) {
  int amplification = 100 / currentRange;
  rawPercent *= amplification;
  rawPercent = constrain(rawPercent, 0, 100);
}
```

3.7 FIRMWARE PERFORMANCE AND OPTIMISATION

MAUDLIN's performance on the Arduino Pro Mini (16MHz) has been tested in situ to meet the demands of real-time Eurorack and music performance, with a focus on trigger rates up to ~250 Hz[3], aligning with the predicted speeds. The 5ms sampling interval, tailored to the ADC's conversion time [4], supports rapid random walk updates, processed in a non-blocking loop to minimise jitter. The MCP4725 DAC's I2C write (~100μs), deferred with a non-blocking flag, contributes to an estimated trigger response of approximately 130μs, as inferred from previous oscilloscope tests across noise ranges (0-15 to 0-260 ADC units). This response enables a maximum trigger rate of ~250 Hz (4 ms period), suitable for both dynamic live modulation and rhythmic applications. In real-world use, MAUDLIN has been informally tested in casual studio sessions and small performances, where the 50ms OLED update interval provided clear visual feedback on step size, range, trigger status, and history without any noticeable lag. Estimated performance under continuous triggering at 250 Hz suggests stable operation, with perturbation calculations (e.g., noise offset for steps) adding a negligible ~2-3 μs (estimated) to the overhead, well within a 4 ms cycle. Field observations in varied EMF environments (e.g., varied indoor settings) indicate that hybrid randomisation adapts effectively.

4 ALTERNATE FIRMWARE MODES

To demonstrate MAUDLIN's versatility as a module with scope beyond the primary application discussed in this paper, several alternate firmware modes are currently under de velopment, utilising the same hardware for various creative tasks. For instance, a quantiser mode maps input CV to selectable scales (e.g. chromatic, major, minor, pentatonic,

3. See Video Reference 7: [Harding, 2025b]
4. ADC conversion time for the ATmega328P is approximately 104 μs at a 16MHz clock with a 128 prescaler, as specified in the datasheet [Atmel Corporation, 2020]

22-tone equal temperament) with CV control over root-note, note trigger and scale form, display ing input sliders and output history. A digital LFO, LFOyeah, produces 12 user-created waveforms at CV-controlled rates and shapes, ideal for modulation purposes. Finally, a Random Source mode samples noise on trigger, normalising to full DAC range for direct output, with graphical sliders and scrolling history. These modes, while in early stage development at the time of writing, hope to underscore the module's adaptability, enabling users to re-purpose it for CV generation, intonation, modulation, or random value genera tion by simply changing the firmware, all applications aim to maintain the low-latency and open-source extensibility aspects and will be delivered via the code repository.

5 OPEN-SOURCE APPROACH

The firmware, written in C++ for Arduino via Platformio IDE[5] along with hardware design files are presented open source [Harding, 2025a] to encourage community use and contributions, aligning with the ubimus emphasis on collaborative creativity. Users may modify or extend functionality, such as adding alternate firmware. The universal nature of CV and Gate inputs, along with 0 - 5 VDC CV output, enables broader applications in modular synthesis. The hardware development presented through this work has potential for greater scope and future development beyond the Eurorack format and towards standalone applications. At the time of writing, multiple alternative

[FormattingFormatting]

5. https://platformio.org/

firmware applications are in development and will be located within the aforementioned code repository once completed. [Harding, 2025a]

6 MUSICAL APPLICATIONS AND UBIMUS RELEVANCE

MAUDLIN has the potential to enhance ubimus practices by providing a hardware-based tool for generative music that is relatively low-cost, accessible, and adaptable. The current random-walk algorithm, seeded and augmented by ambient electrical EMF noise, aids in production of evolving patterns which are constrained within user defined boundaries. These parameters can be set manually via potentiometers and/or modulated in real time via Control Voltage inputs for Step Size, Minimum, and Maximum boundaries. The organic but controlled unpredictability contrasts with deterministic digital systems, enriching the sonic palette for musical applications. The tactile interface and causally linked visual feedback enable musicians to explore randomness intuitively. [Keller et al., 2014].

We contend that constrained randomness of this nature significantly reduces barriers to entry for musical creation, allowing novices to produce musically compelling results without requiring extensive technical or musical expertise [Kramann, 2020]. This approach aligns with the principles of ubimus, which emphasise accessible and ecologically grounded creative frameworks [Keller et al., 2014]. This development aims to facilitate the creation of harmonic structures that evolve over time, offering musically engaging outputs that are accessible to users with minimal training.

Kramann’s work provides some parallels, illustrating how structured, game-like systems with constrained randomness can guide lay participants in musical improvisation [Kramann, 2020]. With a single trigger input, and when coupled with a following quanti sation module, a constrained random walk can be used to generate musically coherent CV pitch sequences by producing complete melodic passages for a VCO, with OLED feedback for accessibility; this mirrors Kramann’s structured improvisation by empowering novices to create engaging music through simple game-like interactions.

The principles of exploration take inspiration from the nonlinear interactions implicitly seen in the Buchla-Serge synthesis paradigm, supporting the idea of generative systems that produce unpredictable yet musically meaningful outcomes, as seen in modules like the Serge Dual Universal Slope Generator [Barton and Fritz, 2024] or Smoothed Stepped Generator [Serge Modular Music Systems,] which by their very nature encourage emergent sonic outcomes.

The open source design invites community-driven enhancements and alternative future applications, such as generative quantisation to 22-tone equal temperament for exam ple [Erlich, 1998], while integration with networked platforms like the Internet of Musical Things [Turchet et al., 2018], may expand its accessibility in the future. Using low-cost embedded hardware, this has the capacity to serve as a model for future embedded systems, with potential applications in generative art, interactive installations, or educational tools [Keller and Lazzarini, 2017] and [Lazzarini and Timoney, 2020]. This democratisation of advanced music technologies supports the vision of collaborative, everyday musical creativity.

6.1 COMPARISON WITH TRADITIONAL EURORACK PATCHING

It is challenging to implement a completely equivalent patch using a series of standard Eu rorack modules; however, as a suggested approximation to this goal and with a comparative range of main features, it would require as a minimum:

- Make Noise Wogglebug (10 HP)[6] – Provides noise, random CV (for steps), and clocked randomness.

- Intellijel Quadrax (14HP)[7] – Handles Sample and Hold on trigger, gate input, and can generate variable steps via envelope modulation.

- Doepfer A-172 Min/Max (4 HP)[8] – Analog min/max selector to clamp the output. Output directly as CV.

[Indent]This approach would require significantly more horizontal width (HP) than the current development. As an approximation; 28 HP. This solution would not provide the user with any form of intuitive visual feedback and is a costly and bulky solution. We would argue that the visual feedback component is vital for the user in this case. The 8 HP design with integrated OLED feedback is more compact and accessible, aligning with the expectations of the Eurorack standard, where module horizontal width (HP) is a key [FormattingFormatting]

6. https://www.makenoisemusic.com/modules/wogglebug/
7. https://intellijel.com/shop/eurorack/quadrax/
8. https://www.doepfer.de/a100_man/A172_man.pdf

factor in determining commercial acceptance whilst maintaining the mobility of the system overall.

7 VIDEO DEMONSTRATIONS

A video playlist showcasing the core and extended functionality of MAUDLIN has been provided to expand on the technical descriptions provided in this extended paper and to provide further detail the practical applications of this development for musical uses. These demonstrations highlight the core functionality, accessibility, and musical usage in the context of Eurorack synthesis, and in the future this playlist will serve as a placeholder for demonstrations of alternate firmware modes as they are publicly released. These video references playlist can be found at the following location: [Harding, 2025b].

8 CONCLUSION AND FUTURE WORK

We believe that MAUDLIN represents a significant contribution to both the ubiquitous music community and the Eurorack synthesis ecosystem by merging analog randomness with digital control and GUI in a compact, low-cost

9 ACKNOWLEDGEMENTS

This work is supported by Ulster University, DLD Campus, N.Ireland.

REFERENCES

Arduino (2025). Arduino pro mini. https://www.arduino.cc/en/Main/ArduinoBoardProMini. Accessed: 2025-05-09.

Atmel Corporation (2020). 8-bit AVR Mi crocontroller with 32K Bytes In-System Programmable Flash: AT mega328/P Datasheet. Microchip Technology Inc. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_ Datasheet.pdf.

Barton, T. and Fritz, I. (2024). Randomness and uncertainty in sound design. In Filimowicz, M., editor, The Routledge Handbook of Sound Design. Routledge.

Cycling ’74 (2025). Drunk reference. https://docs.cycling74.com/max8/refpages/drunk. Max 8 Documentation, Accessed: 2025-05-09.

Doepfer Musikelektronik (2023). Technical details a 100: Construction details. https://doepfer.de/a100_man/a100m_e.htm. Accessed: 2025-18-08.

Erlich, P. (1998). Tuning, tonality, and twenty-two tone temperament. Xenharmonikôn, 17. Revised 4/2/02, An Informal Journal of Experimental Music.

Harding, J. (2025a). Maudlin repository. https://bitbucket.org/Nonzerojohn/maudlin/src/main/. Bitbucket Repository, Accessed: 2025-09-05.

Harding, J. (2025b). Maudlin video repos itory. https://www.youtube.com/playlist?list=PLvcKlQ9XRG3WAhMlwhTqmWXe6nu7HDsR4. Video Repository, Accessed: 2025-11-08.

Keller, D. et al. (2014). Ubiquitous music: Perspectives and chal lenges. Journal of New Music Research, 43(1):1–6.

Knuth, D. E. (1998). Seminumerical algorithms. In The Art of Computer Programming, pages 10–26. Addison-Wesley, third edition.

Konstantopoulos, T. (2009). Markov chains and random walks. https://www2.math.uu.se/~takis/L/McRw/mcrw.pdf. Lecture Notes, Accessed: 2025-11-08.

Kramann, G. (2020). Of renouncing to do something grandiose. In Stolfi, A., Costalonga, L., Messina, M., Keller, D., and Aliel, L., editors, Proceedings of the 10th Workshop on Ubiquitous Music (UbiMus 2020), pages 21–35, Porto Seguro, BA. UFSB.

Lazzarini, V. and Timoney, J. (2020). The analogue com puter as a musical instrument. In Ubiquitous Music. Taylor & Francis.

Microchip Technology Inc. (2009). Mcp4725 12-bit digital-to-analog converter with eeprom memory in sot-23-6. Accessed: 2025-05-09.

Moog Music Inc. (1978). Minimoog service manual. https://www.synthfool.com/docs/Moog/minimoog/Minimoog_Service_Manual.pdf. Figure 9-6, page 100, Accessed: 2025-11-08.

Serge Modular Music Systems. Smooth & stepped function generator. https://sdiy.info/wiki/CGS_Serge_smooth_and_stepped_generator. Accessed: 2025-08-11

Solomon Systech Limited (2012). Ssd1306 128 x 64 dot matrix oled/pled segment/common driver with controller. https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf. Rev 1.1, Accessed: 2025-05-09.

Turchet, L., Fischione, C., Essl, G., Keller, D., and Barthet, M. (2018). Internet of musical things: Vision and challenges. IEEE Access, 6:61994–62017.