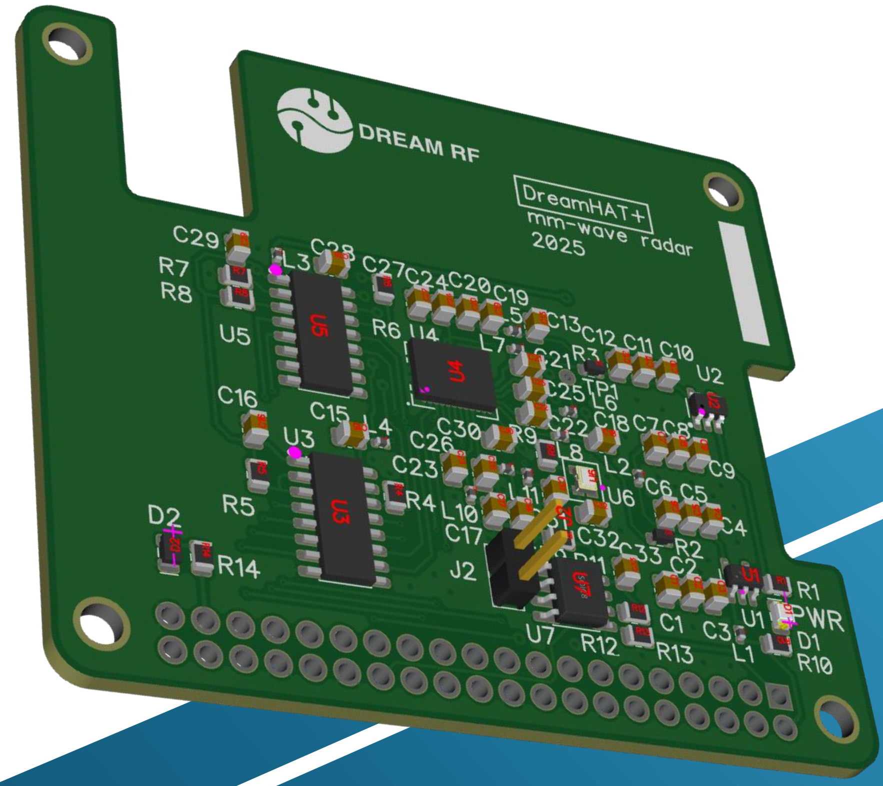
DREAM RF

DreamHat+ Radar

Setup Guide

60 GHz mm-Wave Radar HAT+
For Raspberry Pi 4B and 5

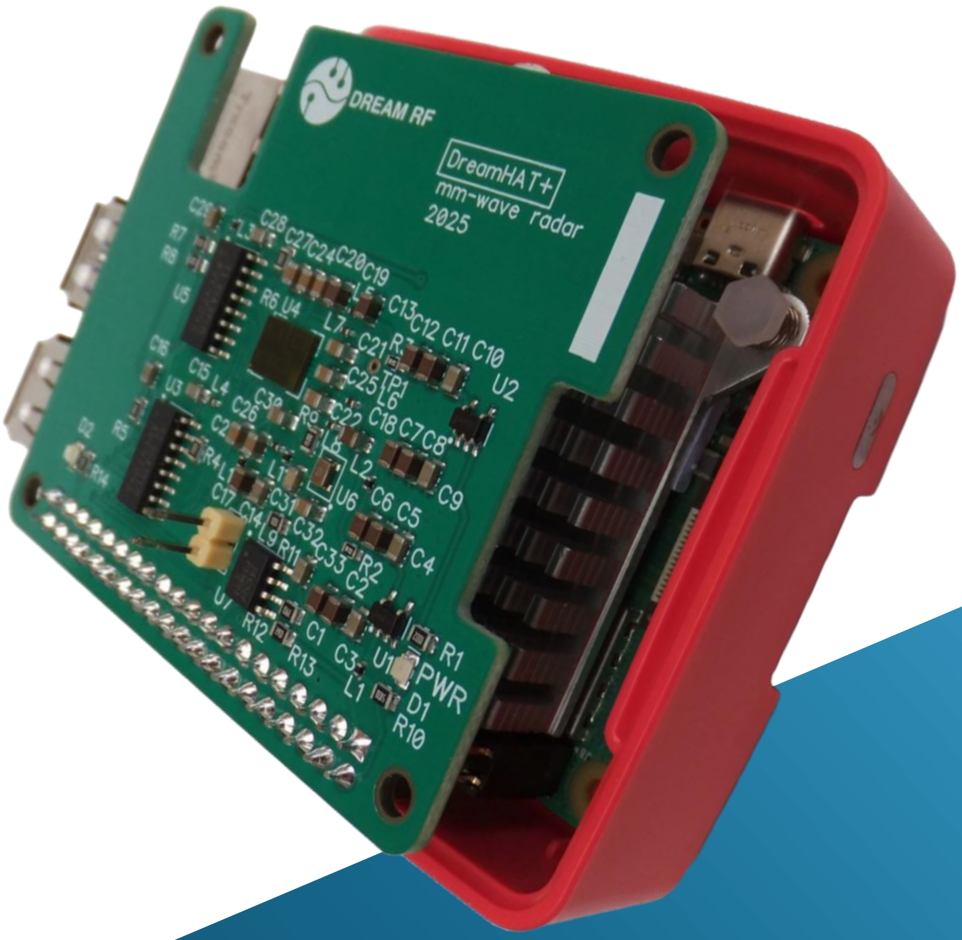


www.dreamboards.co.uk

Published May 2025
Dream RF Ltd

Contents

Welcome to the World of mm-Wave Radar

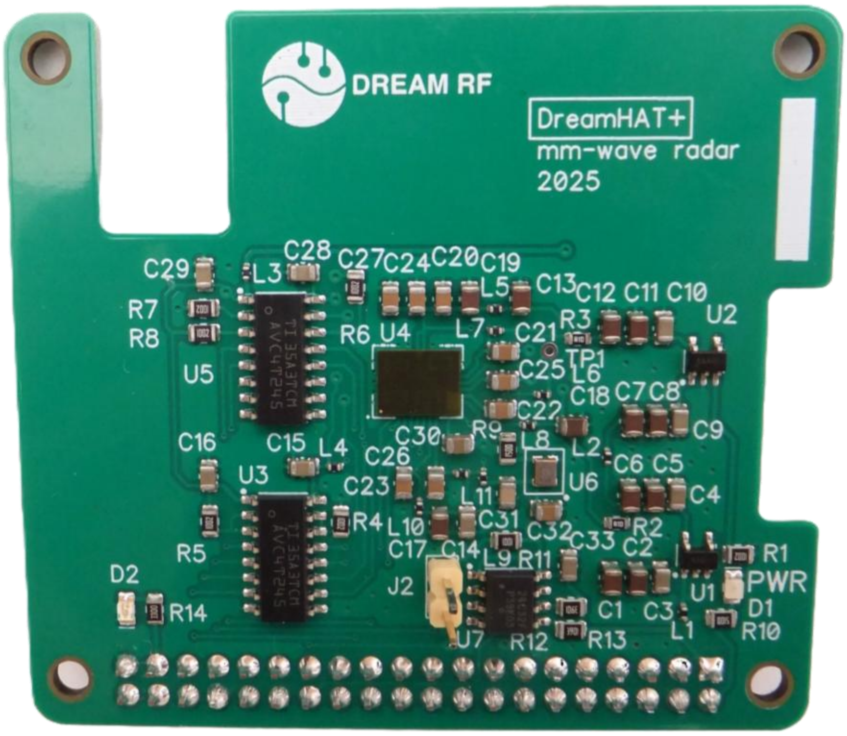


What is mm-Wave Radar?

Millimeter-wave (mm-Wave) radar operates in the electromagnetic spectrum at frequencies between 30 GHz and 300 GHz, corresponding to wavelengths in the millimetre range (1–10 mm). It brings high-precision sensing and reliability to a wide range of applications.

With centimetre-level accuracy and exceptional sensitivity, mm-Wave radar can detect and track objects even in environments where traditional sensors—such as cameras and infrared systems—struggle to perform. It can also penetrate through some materials like plastic, drywall, and clothing,

These advantages have made mm-Wave radar a go-to solution in industries ranging from automotive and robotics to industrial automation and consumer electronics. Whether it's for enhancing safety, improving automation, or exploring innovative applications, mm-Wave radar unlocks new possibilities for intelligent sensing.



The DreamHAT+

The **60 GHz DreamHAT+ mm-wave radar** is a user-friendly mm-Wave HAT+ board that seamlessly integrates with the Raspberry Pi ecosystem.

Our goal is to make mm-Wave radar sensing more accessible, empowering a wider audience to explore and innovate with this powerful technology. Whether you're an educator, a maker, or an engineer, the DreamHAT+ unlocks new possibilities for intelligent sensing in a way that's intuitive and affordable.

Getting Started with the DreamHAT+

Setting up your Raspberry Pi with the provided image is quick and easy. Just follow these steps:

1. Prepare the Raspberry Pi Image

- **Download** the pre-configured Raspberry Pi image (*mmw-hat.zip*) from the DreamRF GitHub page:

  https://github.com/DreamRF/mm-wave-DreamHat-radar

- **Unzip** the downloaded file using 7-Zip.
- **Write** the image to a microSD card using suitable disk imaging software such as the Raspberry Pi Imager: https://www.raspberrypi.com/news/raspberry-pi-imager-imaging-utility/

2. Boot Up the Raspberry Pi

- Insert the microSD card into your Raspberry Pi.
- Connect a keyboard, mouse, and display.
- Power on the Raspberry Pi.

Note 1**:** On first boot, the system may automatically reboot—this is normal! It’s expanding the file system to utilize the entire microSD card.

Note 2: The default login credentials for the provided Raspberry Pi image are:

Username: **pi**

Password: **MMW-HAT**

It is recommended that you change the default password.

3. Optimize Display Settings

- Set the screen resolution to 1920×1080 (1080p) for a smoother and faster GUI experience.

You are now ready to run the real-world radar examples provided with the DreamHAT+

Real-World Radar Examples

After booting your Raspberry Pi with the *mmw-hat.zip* image, you will be presented with 3 ready-to-go examples (and a readme file) on the desktop:

- **example_1_range_doppler.sh**
- **example_2_tracking.sh**
- **example_3_azimuth.sh**

To run any example, simply double-click any one of the files, then choose "*execute in terminal*".

Each example is described in more detail in the following pages of this Setup Guide. The source code for all examples can also be found in: */home/pi/Documents/MMW-HAT-Release*

Additionally, you can watch the examples being deployed in real-time on DreamRF's YouTube page: https://www.youtube.com/@DreamRF

Each example will generate real-time radar images at c. 5-10 Hz refresh rate. These should be more than enough to get you started and on the way to creating your own unique 60 GHz radar project!

Example 1:
Range-Doppler Plot
(The Classic Radar Visualisation)

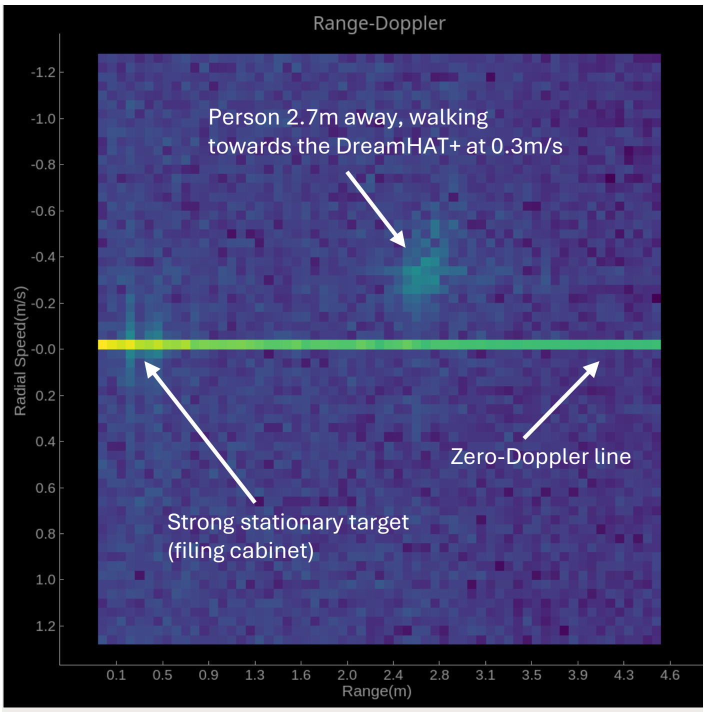
A Range-Doppler plot is an intuitive ways to visualise what the DreamHAT+ radar system detects. It provides two key pieces of information:

x-axis: How far people or objects are from the radar (range)

Y-axis: How fast they are moving relative to the radar (Doppler shift/speed)

At the centre of the plot is the **zero-Doppler line** (a crucial reference point), which represents all the stationary objects that are detected by the radar and includes walls, furniture and static objects.

The brightness of the signal at zero-Doppler indicates how strongly these objects reflect the radar waves.



Moving Target Detection

The real power of the Range-Doppler plot comes when objects move, causing their radar response to shift away from the zero-Doppler line.

Movements toward the DreamHAT+ radar will appear in the positive section above the zero-Doppler line whilst movements away from the radar will appear in the negative section. The distance above or below zero-Doppler line indicates speed—the farther the target response is from the line, the faster the object is moving. For example, a walking person may appear closer to the zero Doppler line, while a running person or rapid hand movement will generate a response further out. The Range-Doppler plot can distinguish and display multiple moving targets simultaneously.

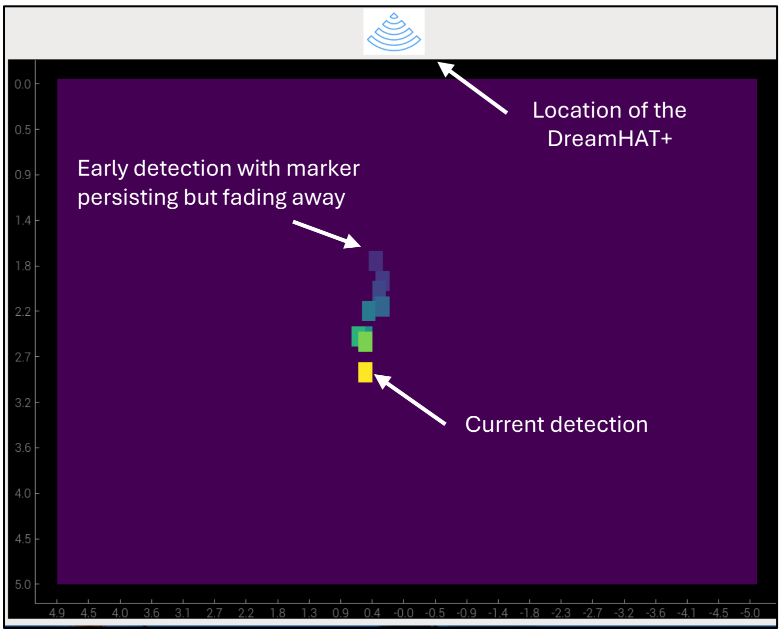
Video Examples

You can see various examples of target detections from the DreamHAT+ radar in a range-Doppler plot on DreamRF's YouTube channel: https://www.youtube.com/@DreamRF

Example 2:
Cartesian (x,y) Plot
(with Tracking and Persistence)

This example turns the DreamHAT+ radar's detections into a dynamic 2D map of its field of view, making it easy to visualize movement.

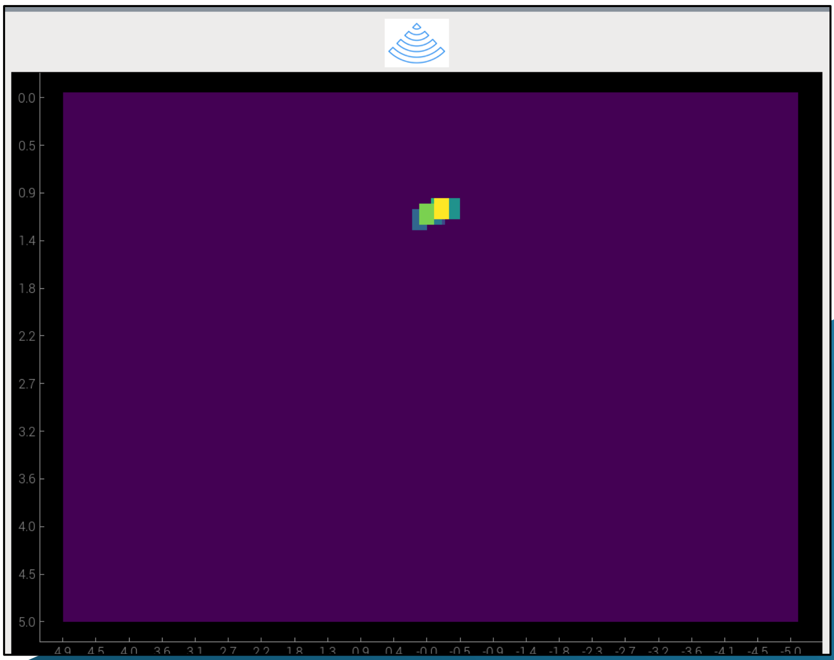
When the radar detects a target, a bright yellow marker appears at its location. As time passes, the marker fades from yellow to green to blue before disappearing, leaving a trail that shows the target's path and speed. This fading effect creates a history of movement, helping you see where the object or person of interest was and how fast it moved—just like a motion trail in a video game!



Radar detection trail for a person walking away from the DreamHAT+ radar

Example 2 also highlights the high sensitivity of DreamHAT+ radar.

Even when a person tries their absolute best to stand completely still, the radar still detects tiny, involuntary movements—like breathing, slight sways, or subtle shifts. On the plot, these tiny motions cause small fluctuations in the marker positions, demonstrating that the radar can pick up movement we don't even notice ourselves!



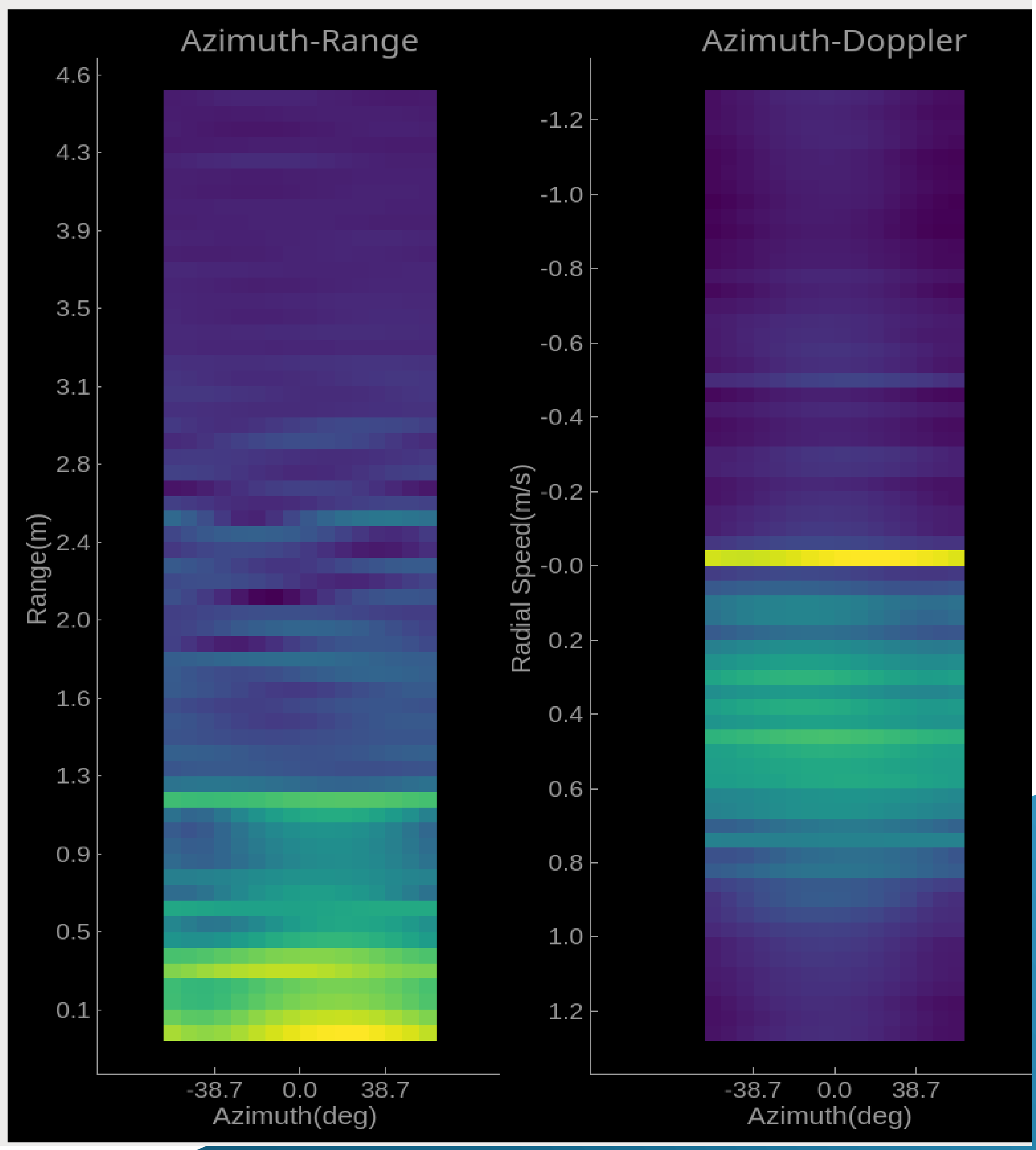
Detection markers for a person attempting to stand completely still

Example 3:
Azimuth-Range and -Doppler Plots

The DreamHAT+ is equipped with an advanced radar chip that features an L-shaped antenna array configuration. This design enables the radar to measure angles in both the vertical direction (elevation) and the horizontal direction (azimuth). As a result, the radar can detect and track movement across multiple directions rather than just in a single plane.

In Example 3, this capability is demonstrated through two types of real-time data visualizations:

- **Azimuth-Range Plot** – Provides a top-down view of detected objects by showing their position relative to the radar.
- **Azimuth-Doppler Plot** – Represents movement relative to the radar.



Example 4:
Data Capture for Offline Processing

The DreamHAT+ offers built-in functionality to effortlessly capture and save raw radar data, giving you the freedom to apply custom signal processing techniques and machine learning algorithms offline.

This flexibility empowers you to develop innovative radar applications, unlocking new possibilities in sensing and analysis.

The **example_4_offline_processing** folder located in */Documents/ MMW-HAT-Release* contains:

- ***data_collection.py*** - A script to record and save the radar data.
- ***offline_processing.py*** **-** A script to process and visualise the data.
- A '*Data*' folder where the recorded data files are stored .
- A *readme* file with additional details.

Step 1: Capture and save the raw radar data

→ Open a terminal window and start the data capture script using:

python data_collection.py

→ To stop the recording , press **CTRL+C** in the terminal.

Note that the longer you record, the larger the saved file will be.

This raw data will be saved as a binary (.bin) file in the data folder in the format:

mmw_spi_YYYYMMDD_HHMMSS_ABC.bin

where: **[YYYMMDD]** is date in year (YYYY), month (MM), and day (DD) format. **[HHMMSS]** is the time in hours (HH), minutes (MM), and seconds (SS) using a 24-hour clock. **[ABC]** is a timestamp recorded in milliseconds for users who wish to synchronise the radar data with other types of sensors.

Step 2: Process the data for visualisation

→ Open ***offline_processing.py*** in a text editor.

Locate the line: *data_fn = "data/mmw_spi_20241119_081114_994.bin"*

→ Replace the example filename "*mmv_spi_20241119_…"* with the actual filename of the data which you have just recorded and saved.

→ From the terminal window run the processing script using **python *offline_processing.py***

A new folder (*mmw_spi*) will be created in */Documents/ MMW-HAT-Release* containing .jpg post-processing images of the recorded radar data: These are:

- Range-Doppler plots
- Azimuth Range plots
- Azimuth-Doppler plots

DREAM RF